

AD-A142 498

RANGE SAFETY REAL-TIME PROGRAM UPDATE ALGORITHM
DEVELOPMENT(U) TYBRIN CORP FORT WALTON BEACH FL
30 APR 82 ER-TC-SER-82-5 SBI-AD-E800 956

1/1

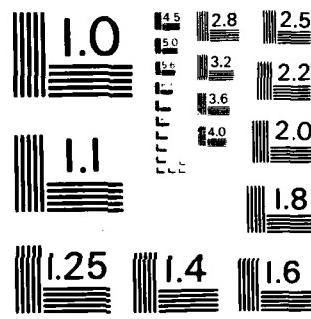
UNCLASSIFIED

F08835-79-C-0140

F/G 12/1

NL

END
DTIC
B-84



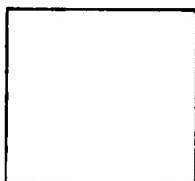
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

AD-A142 498

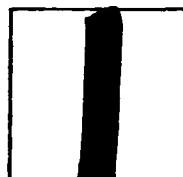
DTIC ACCESSION NUMBER

PHOTOGRAPH THIS SHEET

AD-E800956



LEVEL



INVENTORY

Rpt. No. ER-TC-SER-82-5

DOCUMENT IDENTIFICATION

Contact F08635-79-C-0140

30 Apr '82

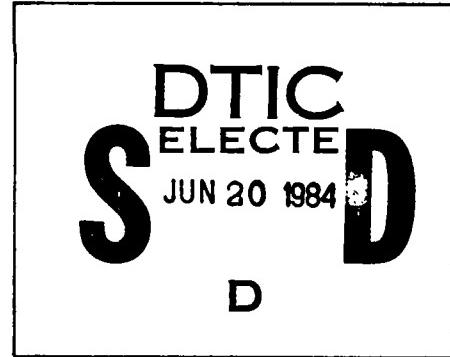
DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR	
NTIS	GRA&I
DTIC	TAB
UNANNOUNCED	
JUSTIFICATION	
BY	
DISTRIBUTION /	
AVAILABILITY CODES	
DIST	AVAIL AND/OR SPECIAL
A/1	

DISTRIBUTION STAMP



DATE RECEIVED IN DTIC

84 06 19 055

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

AD-A142 498

Approved for public release; no
limitation on distribution

Engineering Report ER-TC-SER 82-5

RANGE SAFETY REAL-TIME
PROGRAM UPDATE
ALGORITHM DEVELOPMENT

30 April 1982

Prepared for the
Armament Division (AD)

Directorate of Range Safety (SER)

Contract No. FO8635-79-C-0140
Study Task Order No. SER 82-5

Prepared by
Tybrin Corporation
2018-D Lewis Turner Blvd.
Fort Walton Beach, Florida 32548

TABLE OF CONTENTS

<u>Section No.</u>	<u>Description</u>	<u>Page No.</u>
	List of Figures	iii
	List of Tables	iv
	Foreword	v
	Abstract	vi
1.0	Introduction	1-1
2.0	Summary and Recommendations	2-1
2.1	Summary	2-1
2.2	Recommendations	2-1
3.0	Real-Time Impact Prediction Algorithms	3-1
3.1	Background	3-1
3.2	Lagrangian Interpolation and Extrapolation of Multivariate Table	3-2
3.3	Trajectory Integrator	3-5
3.4	Vehicle and Fragment Weight, Area and Drag Data	3-6
3.5	Small β Impact Prediction Tables	3-7
3.6	Large β Fragment Impact Prediction	3-11
4.0	Real-Time Radar Digital Filter Algorithm	4-1
4.1	Background	4-1
4.2	Recommended Algorithm	4-8
4.2.1	Algorithm Development	4-8
4.2.2	Filter Effective Memory	4-10
4.2.3	Initialization, Reinitialization	4-14
4.2.4	Variance Estimation	4-14
4.2.5	Data Editing	4-15
4.2.6	Filter Adaptation	4-16
4.2.7	Filter Coordinate Systems	4-17
4.2.8	Output Characteristics	4-19
5.0	Real-Time Telemetry Data Processing Algorithms	5-1
5.1	Background	5-1

Table of Contents (Cont'd)

<u>Section No.</u>	<u>Description</u>	<u>Page No.</u>
5.2	Coordinate Systems	5-5
5.3	Recommended Radar/INS Combinative Algorithm	5-6
5.4	Recommended Radar/Altimeter Combinative Algorithm	5-7
	References	R-1
	Appendix A	A-1
	Appendix B	B-1
	Appendix C	C-1

LIST OF FIGURES

<u>Figure No.</u>	<u>Description</u>	<u>Page No.</u>
3.1.1	Debris Triangle/Centerline Geometry	3-3
3.5.1	Macro Flow Chart of Small β Impact Prediction Process	3-10
3.6.1	Macro Flow Diagram of the Proposed Large β Impact Prediction Process	3-16
4.1.1	TARGET Filter Output 32 Bit Word Length	4-2
4.1.2	Exponential Filter Output 32 Bit Word Length	4-2
4.2.1.1	Test Bed Adaptive Exponential Filter Subroutine	4-11
4.2.8.1	Velocity Outputs of the Adaptive Exponential Filter Simulated A-20 Range Data on AMRAAM Test	4-21
4.2.8.2	Residual Outputs of the Adaptive Exponential Filter Simulated A-20 Range Data on AMRAAM Test	4-22
5.1.1	Flight Path C Radar East-West IIP Errors	5-3
5.1.2	Flight Path C INS East-West IIP Errors	5-3
5.1.3	Flight Path C Radar/INS East-West Impact Errors	5-4
5.1.4	Flight Path C Radar/Altimeter East-West Impact Errors	5-4
5.3.1	Conceptual Flow of Radar/INS Combinative Data Process	5-8
5.4.1	Conceptual Flow of Combinative Radar-Altimeter Data Processing	5-10

LIST OF TABLES

<u>Table No.</u>	<u>Description</u>	<u>Page No.</u>
3.6.1	Large β Fragment Impact Predictions (Wind Profile: Constant 100 ft/sec from West)	3-13

FOREWORD

The work in this report was sponsored by the Directorate of Range Safety (SER) at the Armament Division under Contract No. F08635-79-C-0140 and is a deliverable item under Study Task Order SER 82-6. The work was monitored by Mr. R. H. Thompson.

ABSTRACT

Algorithms recommended for inclusion in an update of the AD real-time range safety control program, P2457, are developed. The developed algorithms include ones for large and small beta fragment impact prediction by table look-up procedures, adaptive exponential digital filtering of radar data, and the optimal combining of radar and telemetered navigation system/altimeter system data for use in impact prediction. The algorithms are presented in a form conducive to fast, efficient incorporation in the P2457 update.

1.0 INTRODUCTION

During Task SER 82-1, algorithms which should be included in an updated version of the real-time range safety control program (P2457) for the VAX 11/780 real-time computer configuration were identified. The identified algorithms were of three types:

- a) Algorithms being utilized in P2457, requiring no modifications,
- b) Algorithms utilized in P2457, requiring modifications, and
- c) New algorithms not currently included in P2457.

Among the new algorithms identified and recommended were the following:

- 1) Because vacuum instantaneous impact predictions (IIPs) were found to be inappropriate for most tactical weapon tests and the current P2457 drag/wind IIP algorithm was found to be too slow for multiple weapon tests, the use of table lookup IIP algorithms for the largest and smallest beta fragments of a destroyed vehicle were recommended for inclusion in a P2457 update.
- 2) Because the current radar digital filter algorithm used in P2457 was found to have several deficiencies, including instability, a fast, stable exponential weighted filter algorithm was recommended.

3) Because of radar data degradation at the low elevation angles common on advanced AD weapon tests, the inclusion of algorithms for processing telemetered inertial navigation and altimeter data was recommended.

The purpose of this study was to develop the above three algorithms and present them in a form conducive to fast, efficient incorporation into a P2457 update on the VAX 11/780 configuration. Other required algorithms identified in Task SER 82-1 will be developed later.

2.0 SUMMARY AND RECOMMENDATIONS

2.1 Summary

The following algorithms were developed during this study and presented in a form suitable for efficient incorporation in a P2457 update on the AD VAX 11/780 real-time computer configuration.

- 1) Table lookup algorithms for generating the large and small beta fragment vertices of a debris triangle.
- 2) A stable, fast, adaptive exponential filter algorithm to replace the current unstable radar data filter algorithm of P2457.
- 3) Algorithms for combining radar data with telemetered INS and altimeter data to obtain smooth, unbiased trajectory state vector estimates.

The above developed algorithms can be included in the development of a type B5 computer program specification, as recommended in Task SER 82-1.

2.2 Recommendations

It is recommended that the impact prediction, adaptive exponential filtering, and radar/TM data combinative algorithms developed during this study be incorporated in the planned update of the real-time range safety control program, P2457. The algorithms should be included in a type B5 computer program specification for the P2457 update.

3.0 REAL-TIME IMPACT PREDICTION ALGORITHMS

3.1 Background

Several reasons why the current impact prediction algorithms in P2457 should be replaced were cited in Reference 1. The primary reason given for discarding the vacuum impact predictor was that its predictions have become increasingly unrealistic and too restrictive for advanced tactical weapon tests, which are conducted primarily in the lower regions of the atmosphere. For example, consider the following case, which is highly possible for near-future weapons:

Initial Conditions	Altitude	5000 feet
	Speed	Mach 2.7
	Flight path angle	30° above horizontal
	Fragment ballistic coefficient(β)	500
	True down range impact distance	36,500 feet
	Vacuum predicted down range distance	250,000 feet

It can easily be seen that use of the vacuum impact prediction to determine test boundaries would be unnecessarily restrictive.

The primary reason given in Reference 1 for discarding the current drag and wind corrected impact predictor algorithm was that the required numerical integration is too slow to be used for predicting impacts on tests involving multiple weapons and multiple fragments. Each prediction can consume as much as 200 milliseconds on the CDC 6600, and the time required could be greatly increased on the VAX 11/780, where the use of double precision might also be necessary.

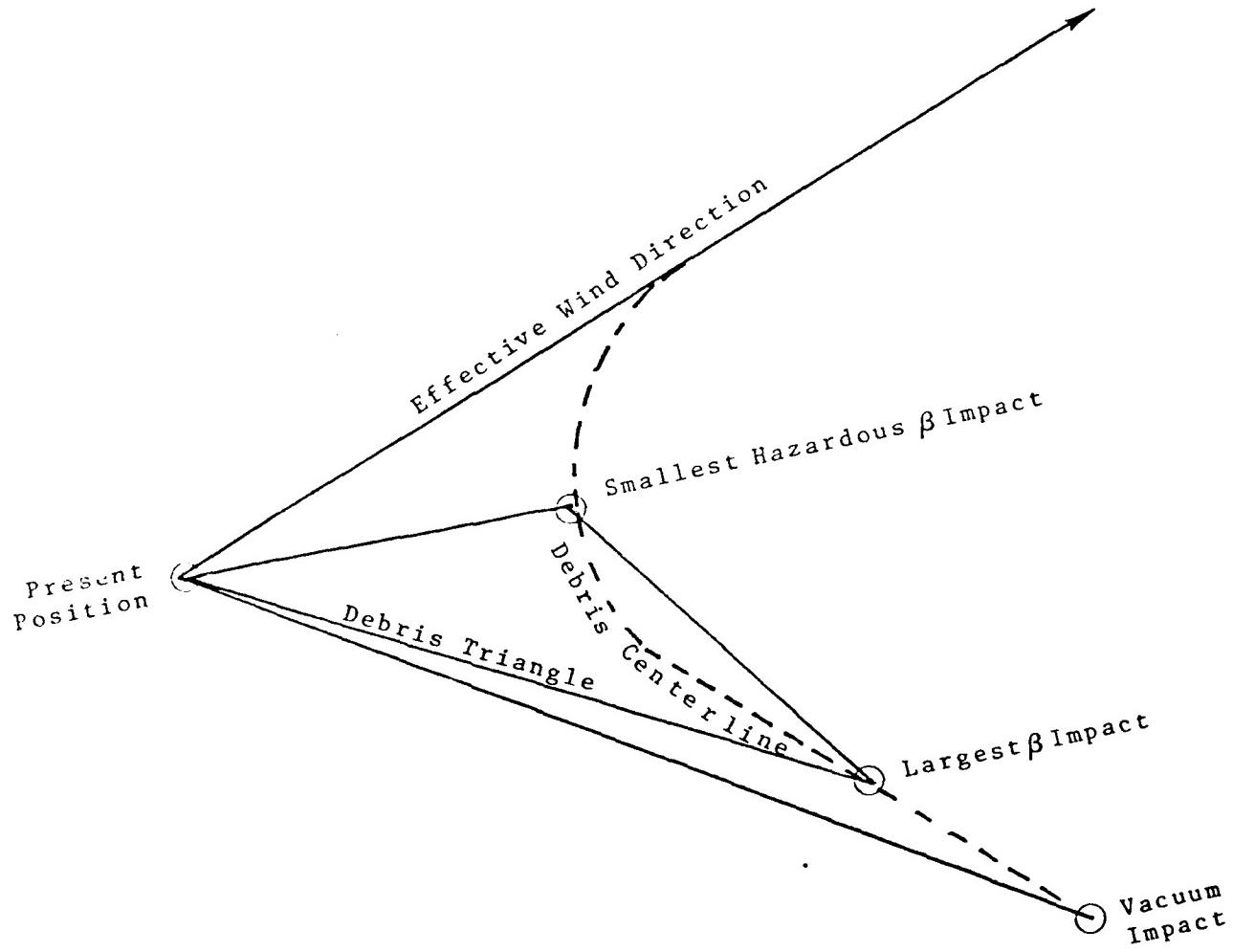
Reference 1 recommended that, for an update of P2457 on the VAX 11/780 configuration, table look-up algorithms similar to those proposed in Reference 2 should be developed. Two types of impact prediction algorithms are needed: a) an algorithm for predicting the impact point of the fragment having the largest β (or the intact vehicle) and b) an algorithm for predicting the impact point of the smallest β fragment which is hazardous. It was also recommended in Reference 1 that the two predicted impact points be presented on a graphics display as two vertices of a debris triangle, as shown in Figure 3.1.1. One of the purposes of this study is to develop the large and small β impact prediction algorithms in a form conducive to inclusion in an update of P2457 on the VAC 11/780 configuration.

3.2 Lagrangian Interpolation and Extrapolation of Multivariate Tables

Table look-up impact predictors require the use of interpolation algorithms. In developing an interpolation algorithm, one must consider the following:

- 1) Number of independent variables
- 2) Spacing of independent variables
- 3) Degree of interpolation polynomial
- 4) Need for extrapolation outside the table

Based upon Reference 2, it is concluded here that the required interpolation algorithm must be able to accept three unevenly spaced independent variables (trivariate interpolation). Selection of the interpolation polynomial degree involves a tradeoff between



DEBRIS TRIANGLE/CENTERLINE GEOMETRY

Figure 3.1.1

using linear interpolation of a table requiring a large block of storage and higher degree interpolation requiring less storage. In order to make optimal use of the 8K byte VAX 11/780 cache memory, the use of small tables would be desirable. Lagrangian interpolation is a widely used method for performing polynominal interpolation and extrapolation of tables having unevenly spaced variables. For second degree polynominal interpolation and extrapolation, the Lagrange formula is:

$$\begin{aligned} f(x) = & ((x-x_1)(x-x_2) / (x_0-x_1)(x_0-x_2)) f(x_0) \\ & + ((x-x_0)(x-x_2) / (x_1-x_0)(x_1-x_2)) f(x_1) \quad (3.2.1) \\ & + ((x-x_0)(x-x_1) / (x_2-x_0)(x_2-x_1)) f(x_2) \end{aligned}$$

where

x = coordinate of desired interpolated point

$f(x)$ = the desired interpolated function approximation

x_0, x_1, x_2 = three values of the independent variable

$f(x_0), f(x_1), f(x_2)$ = functional values for $x=x_0, x_1, x_2$

Available bivariate and univariate second degree Lagrangian interpolation subroutines (References 12 and 13) were combined to form the required trivariate subroutine. A listing of the subroutine (LAGRAN) is provided in Appendix A. Since a primary reason for recommending table look-up drag/wind corrected impact prediction was computational speed, a large three dimensional table ($6 \times 8 \times 10 = 480$ elements) was input to LAGRAN on the CDC 6600 computer and the interpolation was timed. It was found

that 100 interpolations could be performed in 0.045 seconds, or in less than 0.5 milliseconds per interpolation. Thus, the table look-up algorithm is as much as 400 times as fast as the current real-time drag/wind corrected impact predictor (Subroutine IMPACT) and should be suitable for use on tests requiring predictions for several objects.

3.3 Trajectory Integrator

The current drag/wind corrected impact prediction subroutine (IMPACT) was evaluated in Reference 12 and was found to provide satisfactory impact predictions for tactical weapon tests, although minor modifications were recommended. A modified version of IMPACT and its auxiliary subroutines are recommended for inclusion in the P2457 update; however, they will be used to develop the required large and small β impact prediction tables rather than to predict impacts during real-time execution. For the development of the tables the following modifications are recommended:

- 1) The hazardous small β body requiring impact predictions on some tests (e.g., MGD, FIREBOLT) will be the parachute and its load. Because of the slow rate of descent, the integration of parachute trajectories to impact, using the IMPACT subroutine, is extremely time consuming. A version of the fast parachute prediction algorithm presented in Appendix B should be added to the IMPACT codes and be used for generating parachute impact prediction tables.

2) The current version of IMPACT utilizes an exponential atmospheric density model (recently modified from a quadratic model). The densities given by the exponential model can differ from the true densities by several percent and can also result in impact prediction errors of several percent. It is recommended that the subroutine be modified to interpolate atmospheric densities from the latest weather deck. The linear interpolation subroutine INPOL, listed in Appendix B, can be used to perform the required interpolation.

3.4 Vehicle and Fragment Weight, Area and Drag Data

Vehicle or fragment area and weight and coefficients of drag versus Mach number are currently input to P2457 by card prior to a test. For the P2457 update, it is recommended that a library of this information be stored on a tape or disk file and be selected automatically as a function of test vehicle type. One of the recommendations of Task SER 82-6 is that a menu defining the test vehicles be presented on a CRT prior to a test and be selectable by an interactive device such as a joystick controlling a cursor. Among the characteristics defining the vehicle are the following:

- Vehicle name (e.g., Bomarc)
- Shooter/target/missile relationships
- Radar assignments
- Destruct key

If the 82-6 recommendations are accepted, it will be convenient to identify the proper disk file for the vehicle through

use of the selected name. For the purpose of generating the impact prediction tables, it will also be convenient to include on the data files default values defining the vehicle performance envelopes. In order to handle sudden new requirements, the P2457 update should also offer the option of overriding the values on the library file through the keyboard.

3.5 Small β Impact Prediction Tables

The smallest hazardous β fragment impact prediction tables recommended for a P2457 update are based upon the recommendations of Reference 4. Real-time implementation of the algorithm requires the following steps:

- 1) Prior to the test, enter the proposed modified version of IMPACT with the following data:
 - a) Test origin latitude, longitude, height
 - b) Flight azimuth = 0°
 - c) C_D for small β fragment (or reefed and open parachute C_D tables)
 - d) Fragment weight(W)
 - e) Fragment area (A)
 - f) Wind velocity components, speed of sound, and atmospheric density versus altitude
 - g) $X_o = Z_o = V X_o = V Y_o = V Z_o = 0$
 - h) Series of altitudes, h_{oi} $i = 1, n$

- 2) From output of IMPACT, form the impact prediction tables:

Altitude (feet)	X(North) (feet)	Z(East) (feet)
h_1	ΔX_1	ΔZ_1
h_2	ΔX_2	ΔZ_2
.	.	.
.	.	.
.	.	.
h_n	ΔX_n	ΔZ_n

Here, h_n should be higher than the maximum expected altitude during the test.

- 3) During real-time execution, linearly interpolate the ΔX , ΔZ tables to the current altitude, h . Subroutine INPOL, Appendix B, can be used for this purpose.
- 4) Approximate the correction for the small β fragment initial crosswind velocity:

$$\Delta X_c = \Delta R_c \cos (\psi)$$

$$\Delta Z_c = \Delta R_c \sin (\psi)$$

$$\psi = \text{ARCTAN} (\dot{z}_o / \dot{x}_o)$$

$$\Delta R_c = (V_T(h))^2 \cos(\gamma) (1 + (\sin(\gamma)) / 2 + \ln(V_0 / V_T(h))) / g$$

$V_T(h)$ = terminal velocity at altitude h (see Appendix B)

$$V_0 = (\dot{x}_o^2 + \dot{y}_o^2 + \dot{z}_o^2)^{1/2}$$

$$\gamma = \text{ARCSIN}(\dot{y}_o / V_0)$$

$\dot{x}_o, \dot{y}_o, \dot{z}_o$ = current velocity vector components

g = acceleration of gravity (32.2 ft/sec^2)

5) Sum the corrections to obtain the total corrections:

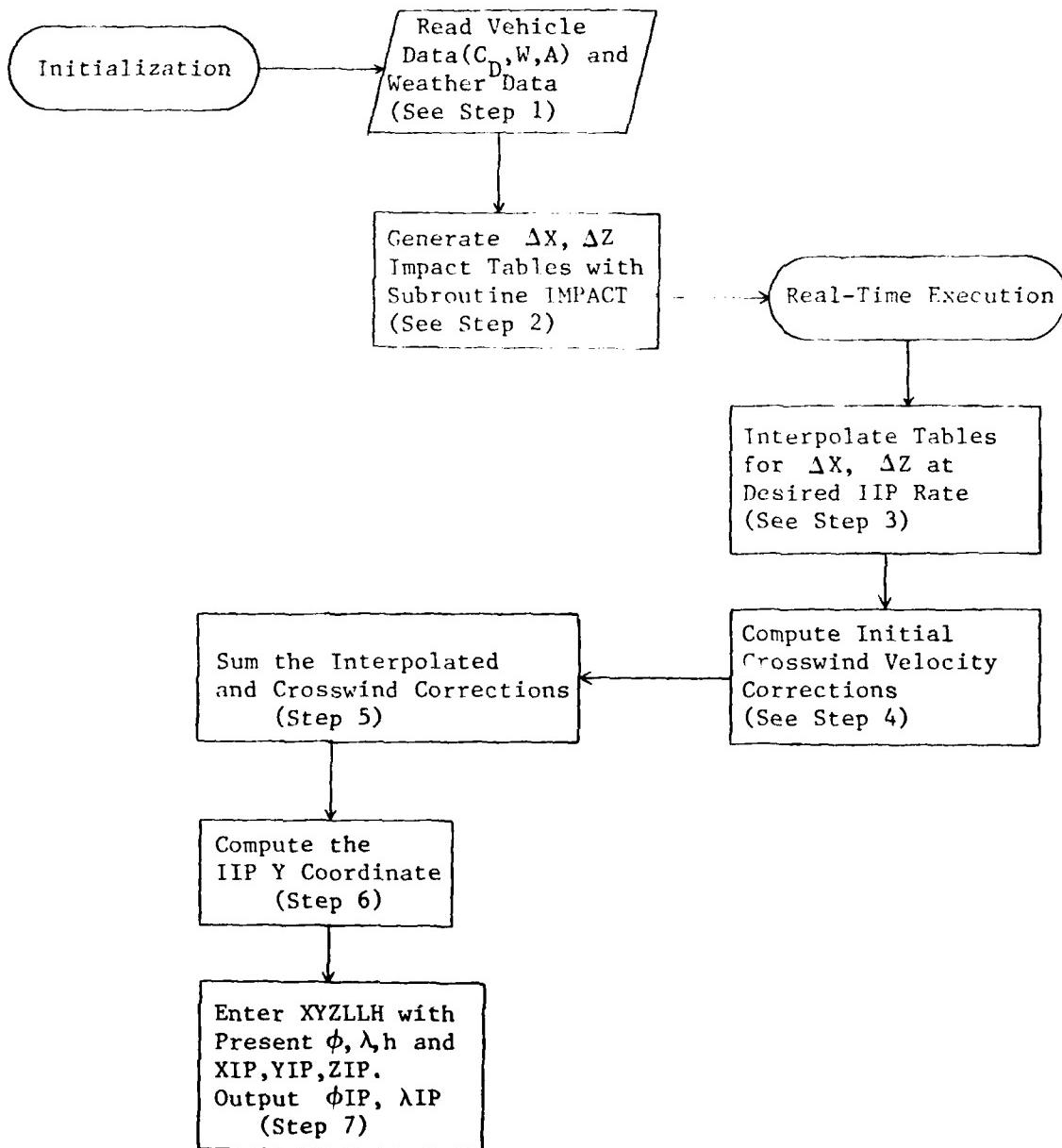
$$\Delta X_{IP} = \Delta X_h + \Delta X_c$$

$$\Delta Z_{IP} = \Delta Z_h + \Delta Z_c$$

6) Approximate ΔY_{IP} : $\Delta Y_{IP} = -((\Delta X_{IP})^2 + (\Delta Z_{IP})^2)/(4.18 \times 10^7)$

7) Enter subroutine XYZLLH with the present position
latitude and longitude and with ΔX_{IP} , ΔY_{IP} , ΔZ_{IP} to
obtain the small β impact prediction.

A macro flow chart of the the proposed small β impact
prediction process is presented in Figure 3.5.1.



Macro Flow Chart of Small β Impact Prediction Process

Figure 3.5.1

3.6 Large β Fragment Impact Prediction

Ideally, the large β downrange, crossrange impact lookup tables should be based upon four independent fragment initial condition variables:

h - altitude

α - flight path angle, positive above horizontal

V - velocity magnitude, or speed

ψ - clockwise heading from north

Using the current estimates of the four parameters from radar data as inputs, the four dimensional tables could be entered in real-time to obtain interpolated values of the downrange and crossrange impact distances.

The inclusion of ψ as one of the independent variables is due mainly to the questionable need to account for wind forces, although small coriolis force effects also vary with the initial heading. In the Reference 3 study of large β impact predictors, the effects of wind on the predictions was considered negligible relative to the large differences between the drag-corrected predictions and the vacuum predictions, and ψ was therefore not included as one of the independent table variables. Analyses conducted during this study showed that the inclusion of ψ would require, for some tests, tables exceeding 1000 storage locations. This was considered excessive relative to the impact prediction improvement which could be achieved; therefore, it is one of the recommendations of this study that

the large β tables should be functions of only altitude, flight path angle and velocity magnitude. For tests in which the nominal trajectory heading is known, most of the error due to wind can still be accounted for by entering this nominal heading into the impact prediction subroutine before generating the tables. Table 3.6.1 shows some "worst case" effects of ignoring the wind in performing large β impact predictions. The cross-range components are due mainly to wind. The tables also show that the vacuum versus drag corrected impact differences are much larger than the errors due to ignoring wind. In generating the tables with the IMPACT subroutine, a 100 ft/sec wind speed from 270° (due west) was imposed.

The steps required in the recommended large β impact prediction procedure are as follows:

1. A record containing the table independent initial condition variables for the test vehicle of interest must be on an input file. Also on the record should be the large β fragment (or intact vehicle) weight, effective drag area and drag versus mach number table. The record should be identified by test vehicle type (e.g., Bomarc). Based upon tests performed during this study, the following numbers of independent variables of each category should be included in the record:

Altitude (feet)	β	V (ft/sec)	α (deg)	ψ (deg)	Downrange IP (feet)	Crossrange IP (feet)	Downrange Vacuum-IP (feet)
5000	500	2000	0	0	20185	1162	35245
30000	500	2000	0	0	44926	3793	86333
50000	500	2000	0	0	69449	4576	111455
5000	500	2000	0	45	21274	812	35245
30000	500	2000	0	45	48146	2685	86333
50000	500	2000	0	45	73220	3270	111455
5000	500	2000	30	0	28269	3098	115637
30000	500	2000	30	0	57094	5477	145896
50000	500	2000	30	0	93244	5839	164290
5000	2000	2000	30	0	61929	2249	115637
30000	2000	2000	30	0	105048	2672	145896
50000	2000	2000	30	0	139249	2368	164290

Large β Fragment Impact Predictions
(Wind Profile: Constant 100 ft/sec from West)

Table 3.6.1

Altitude (feet) - 7 values
Flight path angle (degrees) - 7 values
Velocity magnitude (ft/sec) - 5 values
This will result in a generated table containing 245 dependent variables plus the 19 independent variables. Example of independent variables which might be required for Bomarc:

h	α	V
500	-10	250
5000	0	750
15000	10	1000
30000	30	2000
50000	50	3000
65000	70	
75000	90	

2. If wind corrections are desired, the nominal mean heading, ψ , must be entered from the keyboard for each vehicle requiring impact prediction.
3. Prior to the test, a command to generate the large β impact prediction tables is made. The proper vehicle data will be read from the vehicle data file and all combinations of the independent variables will be used as inputs to the modified IMPACT subroutine. If ψ has not been entered (Step 2 above), wind speed will be set to 0.0 and only one table containing the downrange distances as the dependent variables will be generated (default $\psi = 0^\circ$). If ψ is entered, the available

wind data will be used and an additional table containing the crossrange values will be generated. The tables must be stored in arrays compatible with the LAGRAN subroutine calling sequence (see Appendix A).

4. During real-time execution, after the vehicle present latitude (ϕ), longitude (λ), altitude (h), flight path angle (α), heading (ψ) and velocity magnitude (V) have been determined from radar (or other) data, subroutine LAGRAN will be called for interpolation of the downrange and crossrange impact prediction tables (see Appendix A). The outputs will be X (downrange) and Z (crossrange).
5. Subroutine XYZLLH must be entered with the following data to obtain the desired large β impact point:

Present latitude (ϕ)

Present longitude (λ)

Present altitude (h)

Present heading (ψ)

Interpolated X_{IP}

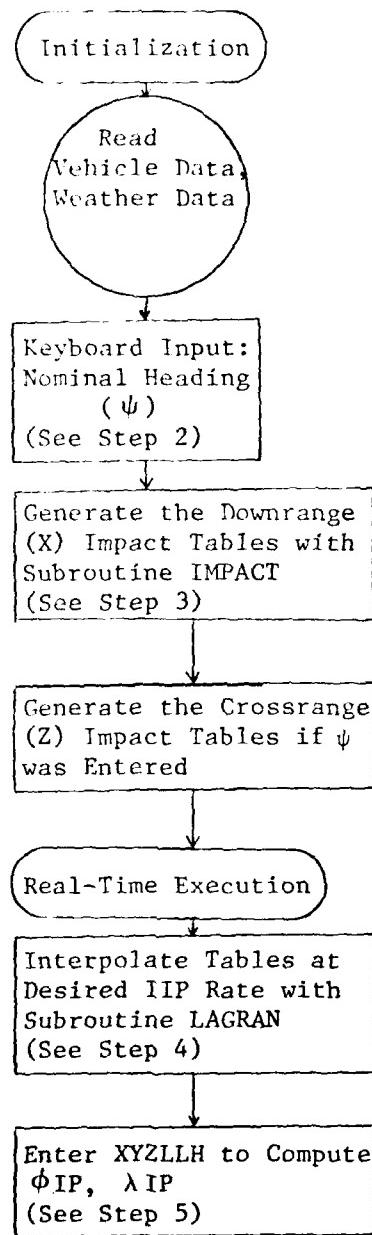
Interpolated Z_{IP}

Computed $Y_{IP} = -(h + (X_{IP}^2 + Z_{IP}^2)/(4.18 \times 10^7))$

KODE = 3

XYZLLH will return ϕ_{IP} , λ_{IP} and h_{IP}

A macro flow chart of the large β fragment impact prediction process is presented in Figure 3.6.1.



Macro Flow Diagram of the Proposed Large β Impact Prediction Process

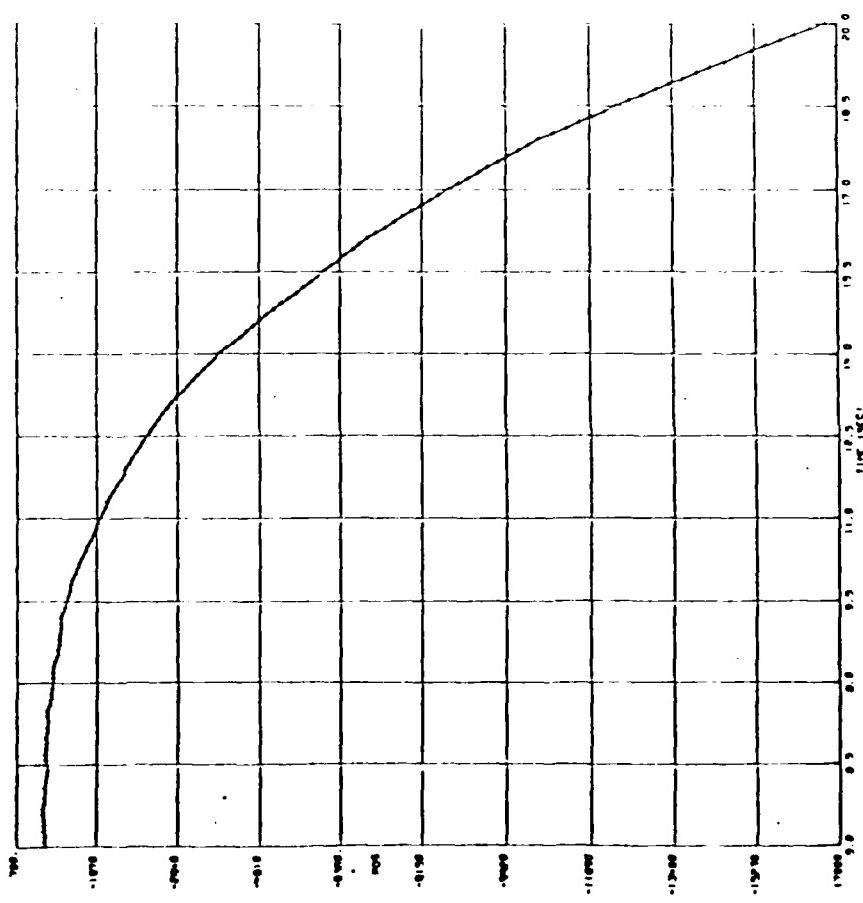
Figure 3.6.1

4.0 REAL-TIME RADAR DIGITAL FILTER ALGORITHM

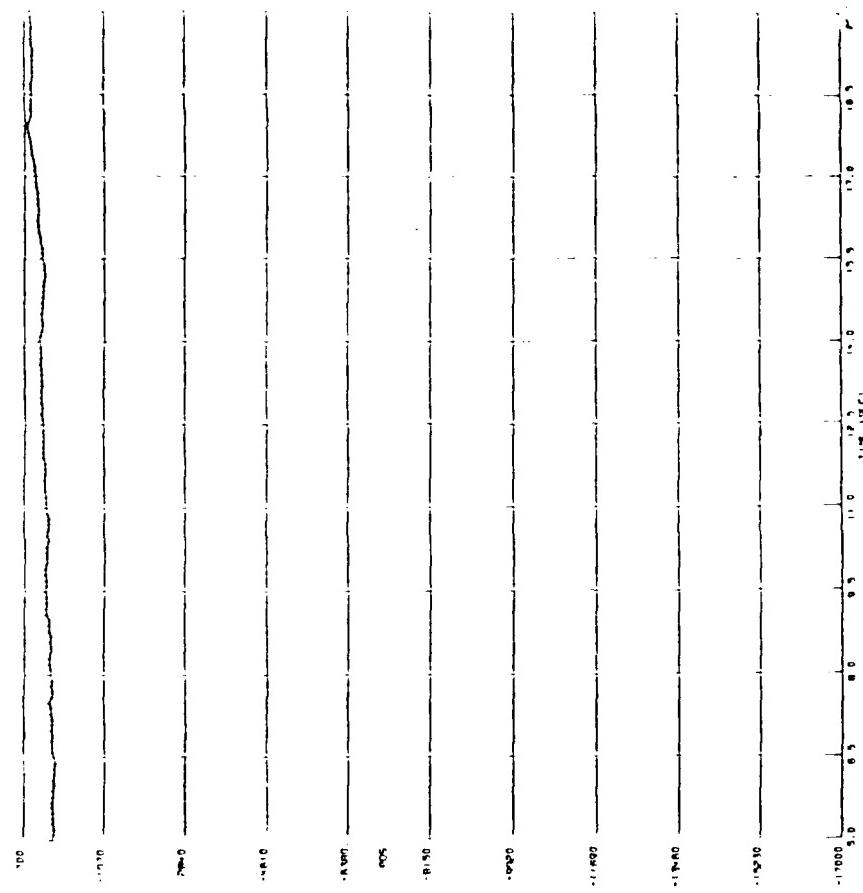
4.1 Background

It was recommended in Reference 1 that the current real-time radar data digital filter algorithm, which is contained in subroutine TARGET of P2457, should be replaced with a filter of the exponentially weighted type if P2457 is updated on the VAX 11/780. Reasons given for this recommendation were the following:

1. The TARGET filter is unstable, being completely unable to eliminate past errors in its output (e.g., errors due to roundoff). Divergence of the filter due to roundoff errors when run on a 32 bit computer is illustrated in Figure 4.1.1. The stable output of an exponential filter, using the same input data, is shown in Figure 4.1.2.
2. The TARGET filter requires a lengthy time period (normally five seconds) to become reinitialized after data dropouts.
3. In order to avoid divergence, the TARGET algorithm would have to be written in double precision for the VAX 11/780. The exponential algorithms would require only single precision.
4. As shown in Reference 9, the TARGET algorithm requires several times as many computational steps and several times the storage of an exponentially



TARGET Filter Output
32 Bit Word Length
Figure 4.1.1



Exponential Filter Output
32 Bit Word Length
Figure 4.1.2

weighted filter. This disadvantage would be increased (by a factor as large as four) because of the need for double precision on the VAX 11/780.

5. The proposed expansion of P2457 to handle up to eight radar inputs will further increase the disadvantages of using the TARGET algorithm, in terms of speed and storage.

The recommendation of an exponential filter algorithm in place of the TARGET algorithm was based largely upon the results of two previous studies (References 8 and 9). In Reference 8, the characteristics of real-time filters which have been used in range safety work at the various test ranges was discussed. All of these filters, in various ways, attempt to compromise between

- (a) reducing the ratio of output noise to input noise to a minimum and
- (b) reducing the bias in the filter output to a minimum.

The method of least squares is a universal approach to achieving (a). Editing schemes involving trend detection with subsequent modification of filter weights are used to achieve (b). Two other important choices often enter into the design of a range safety digital radar filter:

1. A choice of filtering in the radar's natural coordinates (azimuth, elevation, range in the case of the FPS-16) or in a rectangular cartesian coordinate system.

2. A choice of constraining the output of the filter to satisfy a physically realistic set of differential equations (including the forces of drag, lift, thrust, gravitation, etc.) or constraining the output to follow a low degree polynomial.

The choice of the real-time filters in current range safety use at the various ranges has been dictated largely by computer storage and speed requirements. As a result, recursive exponential least squares filtering of cartesian coordinate data with polynomial constraints has been the leading choice. This type of filtering also goes by the names " α , β , γ filtering", "discounted least squares", and "fading memory polynomial filtering". It has the following features.

1. It is fast. For example, the recursive algorithm for a quadratic exponential filter can be expressed as

$$\left. \begin{array}{l} \varepsilon = x^0 - x - \dot{x}T_1 - \ddot{x}T_2 \\ x = x + \alpha\varepsilon \\ \dot{x} = \dot{x} + \dot{x}T_1 + \beta\varepsilon \\ \ddot{x} = \ddot{x} + \gamma\varepsilon \end{array} \right\} \quad (4.1.1)$$

where

x^0 is the current observation,
 T_1 is the time increment between observations,
 $T_2 = .5T_1^2$
 α, β, γ are the filter weights, and
 x, \dot{x}, \ddot{x} is the current state vector estimate.

2. It requires a minimum of computer storage.

Only 10 memory locations are required for the above algorithm.

3. It is readily adaptable. The weights, α , β , and γ can be expressed as simple functions of one constant c . Using trend checks on the residuals (the ϵ 's in the above recursion) as a basis, the filter's effective memory may be easily contracted or expanded to arrive at the desired compromise between noise reduction and bias and stability control.

4. It gives the most weight to the most recent data.

This allows quick response to changes in target motion.

5. It is stable. Due to the fact that weight given past data and past state vector estimates are a function of age, past input and computational errors gradually disappear.

Reference 9 contained a comparison of various test range filters in terms of speed and storage requirements. The speed comparisons were in terms of the number of arithmetical operations required to fit one channel of data (e.g., the X position coordinate) to a second degree polynomial. The comparisons for state vector computation only (no editing, no variance estimation), were as follows:

<u>Algorithm</u>	<u>Number of Operations</u>
Moving Arc (51 points)	306+
Moving Arc (N points)	$6N+$
Current AD TARGET (after initialization)	33
Modified AD TARGET (from Ref.8 - after initialization)	27
Kalman	257
Exponential	13

The numerical storage requirement comparisons, again assuming estimation of position, velocity and acceleration with a second degree polynominal fit to one input channel, were as follows:

<u>Algorithm</u>	<u>Number of Storage Cells</u>
Moving Arc (51 points)	207
Moving Arc (N points)	$4N+3$
Current AD TARGET (51 points - after initialization)	127
Modified AD TARGET (51 points - after initialization)	63
Kalman	64
Exponential	10

Reference 9 mentioned two deficiencies of exponential filters:

- The filters are not self-starting. They need approximate initial state vectors computed by some other method.

- Overshoot and undershoot settling problems are common after sudden changes in the input signal characteristics due to such events as rocket ignition, burnout and stage separation.

Reference 9 showed that the first deficiency could be overcome by using an "expanding memory" filter algorithm and that the second could be overcome by applying multiple exponential filters having different weights to the input data and using an intercomparison scheme to select the optimal outputs. The recommended second degree polynominal expanding memory filter is recursively updated as follows:

Starting with $X = X^0$, $\dot{X} = \ddot{X} = j = 0$,

$$\left. \begin{array}{l} D = (j+3)(j+2)(j+1) \\ X = X + \dot{X}_\tau + \ddot{X}_\tau^2 / 2 \\ E = X^0 - X \\ X = X + 3(3j^2 + 3j + 2)E/D \\ \dot{X} = \dot{X} + \ddot{X}_\tau + 18(2j+1)E/(D_\tau) \\ \ddot{X} = \ddot{X} + 60E/(D_\tau^2) \\ j = j+1 \end{array} \right\} \quad (4.1.2)$$

where

X^0 is the current observation,
 τ is the time between samples,
 j is the sample number ($j=0, 1 \dots n$) and
 \dot{X}, \ddot{X} is the current state vector estimate

4.2 Recommended Algorithm

4.2.1 Algorithm Development

The algorithms defined by Equations 4.1.1 and 4.1.2 are capable of smoothing and differentiating radar data to form an initial position/velocity state vector which can be used for generating real-time instantaneous impact predictions (IIPs). However, in addition to smoothing and differentiating, a range safety digital filter must also perform other important tasks, including:

1. It must edit the input data and replace limited quantities of detected bad data with extrapolated data.
2. It must detect and adapt to sudden changes in target motion or input data quality by optimally modifying the weights assigned the data.
3. It must estimate the uncertainty (variance) in its smoothed position and velocity estimates. These uncertainties can be used for generating IIP uncertainty ellipses and for automatic best data source selection.

The exponential filter algorithm developed during this study is a modified version of the one developed in Reference 9. The previously developed filter had the following features:

1. It accepted data from up to four radars.
2. Filter weights could be selected to give zero, first and second degree polynomial filtering.

3. "On-Track" bit was utilized.
4. Three sets of filter weights were applied to each channel of input.
5. A zero degree polynomial filter was used to estimate noise variance.
6. Adaptation was accomplished through interfilter velocity comparisons, through residual trend detection or by both methods.
7. Limits could be placed upon maximum allowable residual size and upon maximum number of allowable extrapolations.
8. Initialization and reinitialization of the exponential filter was accomplished with the expanding memory least squares polynomial algorithms (Equation 4.1.2 above).
9. Switching from the expanding memory to the exponential algorithms was accomplished through comparison of variance reduction factors (VRFs).
10. Computed variances could be modified with a priori estimates and with computed VRFs.

Most of the above features have been retained in the filter recommended for the P2457 update. The most significant modification recommended is that the required subroutine should accept only one channel at a time. This will allow the subroutine to be more general and will require that calibration corrections, refraction corrections and coordinate transformation

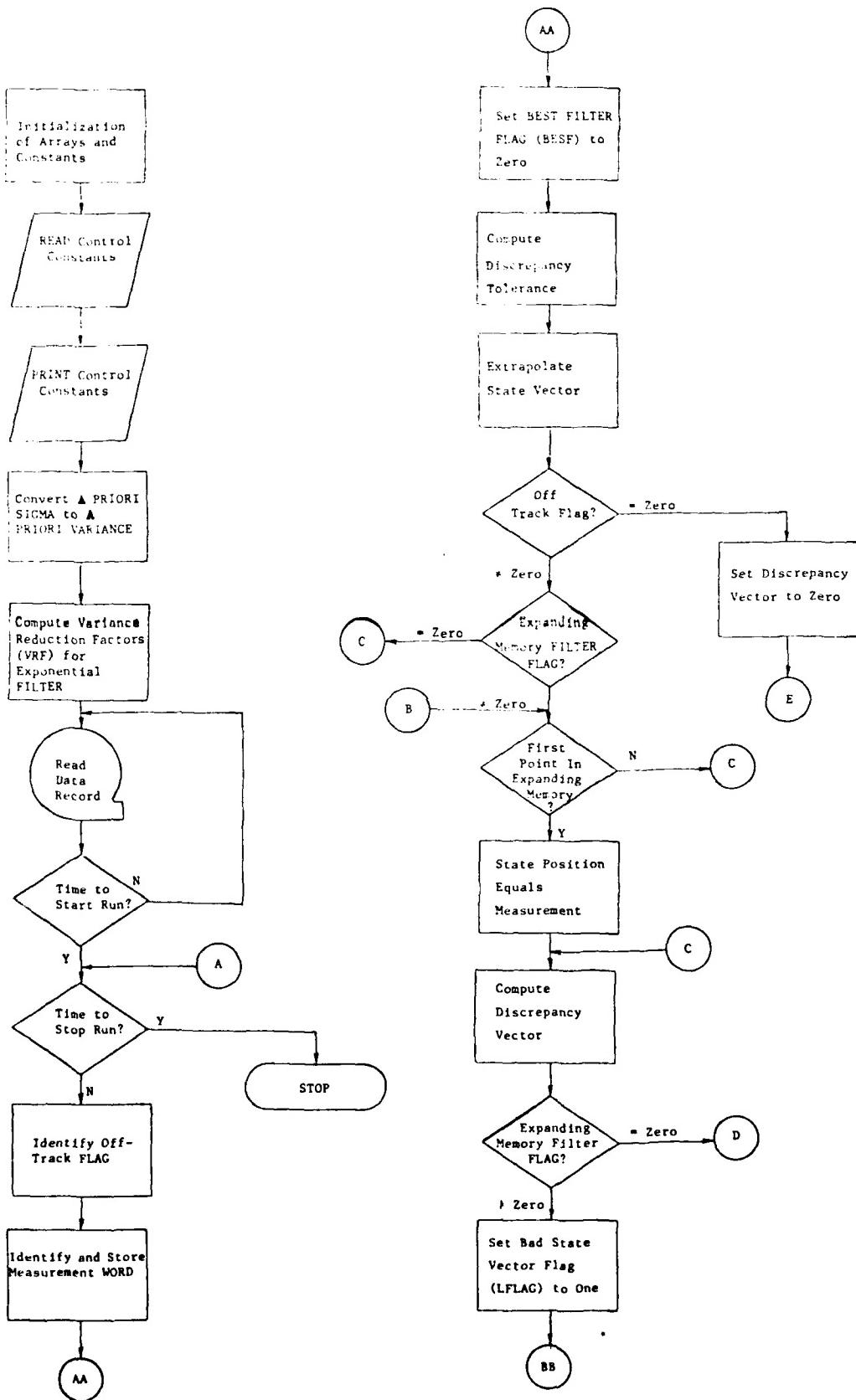
take place in another subroutine. The primary advantage of this recommendation is that the subroutine will be able to accept, without modification, data from systems other than land based, monopulse radars (e.g., telemetered navigation and altimeter system data, MTRACS data). A flow chart of a test bed version of the subroutine, (program RADAR), designed for evaluating the effectiveness of the various options, is presented in Figure 4.2.1.1. A listing of RADAR, a description of input requirements and examples of tabular outputs are provided in Appendix C. A description of the major features of the algorithm follows.

4.2.2 Filter Effective Memory

The Reference 9 exponential filter weights (α, β, γ) were required inputs to the program. In order to make the weighting more meaningful to users of the subroutine, the weighting in RADAR is controlled by inputting the "effective memory" of each filter. The effective memory of an exponential filter is here defined to be the number of points in the span of a moving arc filter which gives the same variance reduction in filtered position data as the exponential filter of the same degree, a standard proposed in Reference 10. The effective memory is used to compute the filter weights as follows:

Given the input filter effective memory, F points,

$$\begin{aligned}\theta &= 1 - 4.36/F \\ \alpha &= 1 - \theta^3 \\ \beta &= 15(1+\theta)(1-\theta)^2 \\ \gamma &= 100(1-\theta)^3\end{aligned}\tag{4.2.2.1}$$



Test Bed Adaptive Exponential Filter Subroutine

Figure 4.2.1.1
4-11

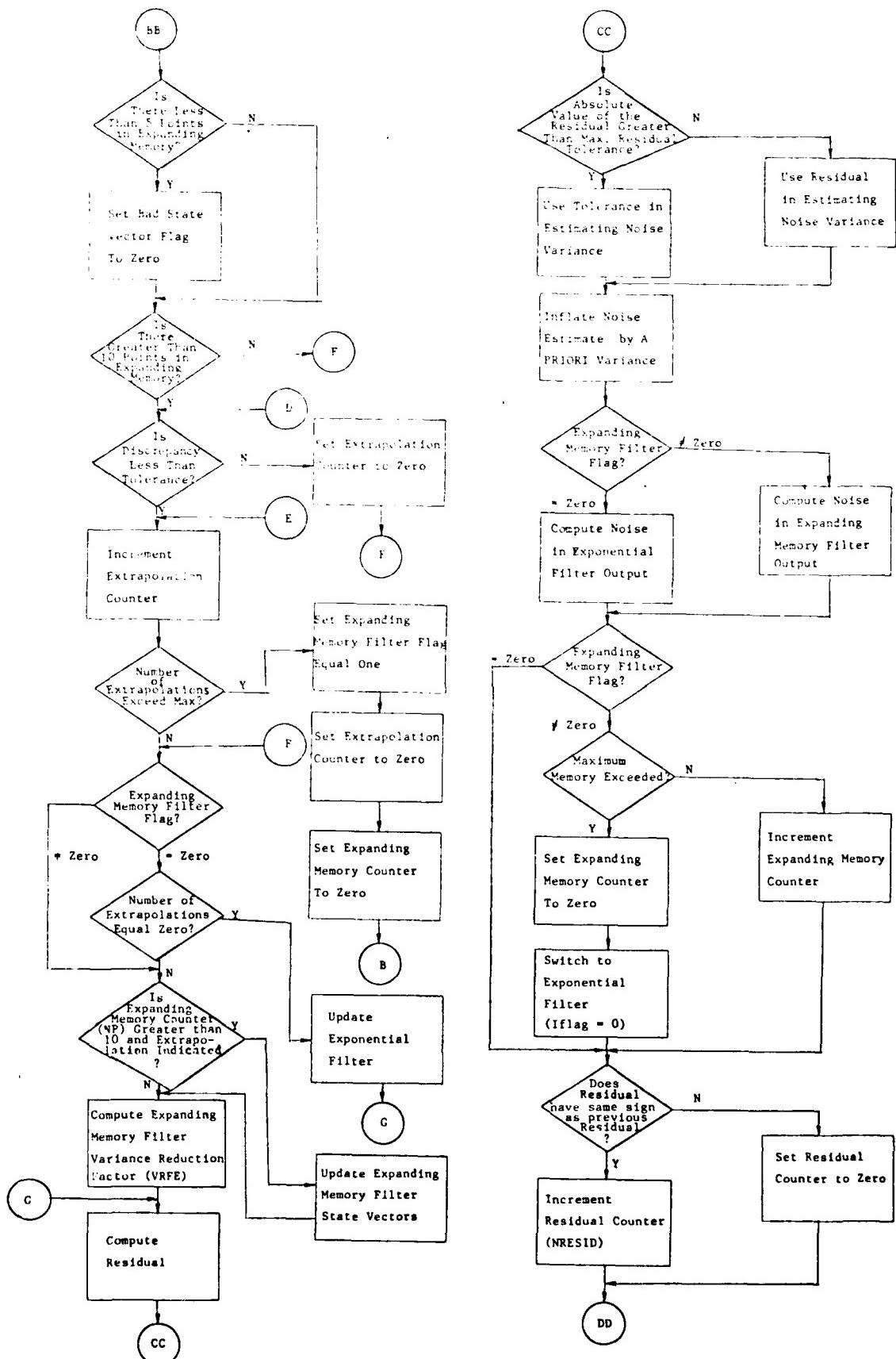


Figure 4.2.1.1 (Cont'd)

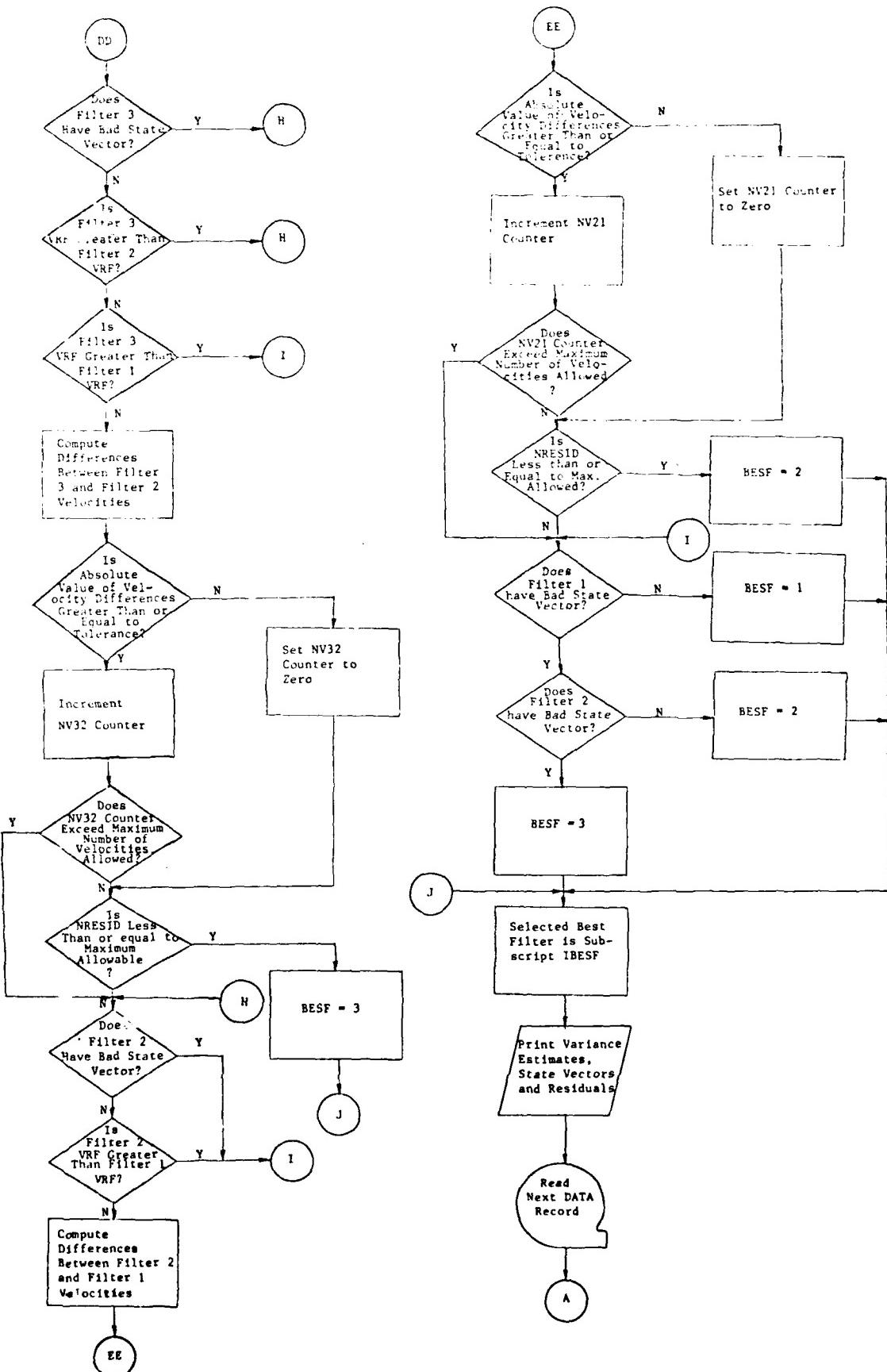


Figure 4.2.1.1 (Cont'd)

The above relationship between the moving arc and exponential filter memories is highly accurate for long memories but only approximate for short memories.

4.2.3 Initialization, Reinitialization

The recommended algorithm for initializing the exponential filter and reinitializing after dropouts was given by Equation 4.1.2, from Reference 15. The initialization algorithm is allowed to expand its memory until its position variance reduction factor (VRF) is less or equal to the position VRF of the desired exponential filter. The filtering is then transferred to the exponential algorithm. The VRFs are computed as follows (from Reference 10):

$$\begin{aligned}\text{Position VRF (Exponential)} &= (1-\theta)(19+24\theta+16\theta^2+6\theta^3 \\ &\quad + \theta^4)/(1+\theta)^5 \\ \text{Velocity VRF (Exponential)} &= 50(1-\theta)^3(49+50\theta+13\theta^2)/(1+\theta)^5 \\ \text{Position VRF (Expanding)} &= (9N^2+27N+24)/(N+1)^3 \\ \text{Velocity VRF (Expanding)} &= (19200N^2+74400N+68400)/(N+3)^5\end{aligned}\tag{4.2.3.1}$$

where N is the number of points in the expanding memory filter span and θ was defined in Equations 4.2.2.1..

4.2.4 Variance Estimation

The recommended noise variance estimation scheme used in RADAR applies a zero degree exponential filter to the squares of the filter residuals. To avoid the use of extremely wild points (which are detected in the editing process), a limit is set upon the maximum residual allowed to be used. The variance estimation algorithms is as follows:

$$V(\text{residual}) = X^{\circ} - X$$

$$\text{If } (|V| > V_{\max}), V = V_{\max} \quad (4.2.4.1)$$

$$S(\text{Variance}) = S + \alpha(V^2 - S)$$

where X° and X were defined in Section 4.1.

The variance estimate output by the filter can be increased by an input a priori value to account for undetectable low frequency noise. The program multiplies the noise variance estimates by the previously defined VRFs (Equation 4.2.3.1) to obtain estimates of variance in the smoothed data.

4.2.5 Data Editing

The recommended data editing scheme used in RADAR compares the input observations against observations computed from an extrapolated state vector. If the differences (REPS) exceed a prescribed value or the offtrack bit is on, the input observation is replaced with the extrapolated value. The algorithm proceeds as follows:

$$\epsilon_3 = \epsilon_1$$

$$\epsilon_4 = 3S^{1/2}$$

$$\text{If } \epsilon_4 > \epsilon_1, \epsilon_3 = \epsilon_4$$

If REPS > ϵ_3 , extrapolated value is used.

Here,

S was defined in Section 4.2.4 and ϵ_1 is an input tolerance value.

An input value, NMAX, places a limit on the maximum number of extrapolations allowed before reinitialization of the filter must take place.

4.2.6 Filter Adaptation

Closely related to data editing is filter adaptation. This consists of varying the filter weights as a function of target dynamics or of data quality. Servo error voltage and AGC measurements have been used for this purpose in the development of real-time digital filters for ON-AXIS radars (Reference 15). In typical usage, if the filtered error voltages exceed prescribed values, the filters are "opened up", giving more weight to newly acquired data and reducing the effective filter memory. AGC measurements are used to place limits on the filter weights. Unfortunately, servo error voltage and AGC measurements are not available at the AD real-time computers; alternative adaptation procedures are required.

Two adaptation tests were developed during the Reference 9 study. These have been modified and extended during this study. The basic purposes of the tests are to:

- a) avoid settling periods after events such as missile stage separation and data dropouts, and
- b) detect possible filter bias or divergence.

The revised adaptation process developed during this study and included in the RADAR filter program makes use of the fact that several exponential filters can be applied in less computer time than is required for other types of filters. This allows the optimum filter to be selected through inter-filter comparisons and simulates the type of adaptation being accomplished

with error voltage and AGC measurements by other ranges. At the same time, it eliminates response and settling problems commonly experienced with other filters. In the adaptation algorithm developed for the Reference 9 study and extended during this study, three exponential filters were applied to each channel of input.

The complete adaptation process is shown in the Figure 4.2.1.1 flow chart, beginning at entry point DD. Several sequential tests must be passed by the filter selected as best. The tests include the following:

- 1) Tests to determine if the filter is extrapolating due to the editing procedure (Section 4.2.5).
The longest filter which is not extrapolating is normally designated best.
- 2) Tests which compare differences between filter outputs against a tolerance value and count the number of successive differences which exceed the tolerance.
A large number of successive differences exceeding the tolerance is assumed to indicate that the longer filter is biased or diverging, and the shorter filter is then designated best. The tolerance (TOL2) and the allowed maximum number of differences (MAXNV) are input values.
- 3) Tests which count the number of successive residuals having the same sign. If the number exceeds an input value (NRMAX), the filter output is assumed to be

biased or diverging, and a shorter filter is selected as best.

4.2.7 Filter Coordinate Systems

Reference 9 details several advantages for filtering in a system's natural coordinate system (e.g., azimuth, elevation, range) rather than in transformed coordinates (e.g., rectangular, cartesian). The advantages include:

- a) Unlike cartesian coordinates, which are subject to cross-coupling of errors, the natural coordinates can usually be considered independent. This simplifies error detection and analysis.
- b) Filter weights can be selected in a more optimal manner. For example, the FPS-16 range channel normally requires a much shorter filter memory than the angle channels.

The Reference 9 study only addressed the use of pulse radar data for range safety control. Since that time, the use of additional systems has been recommended (e.g., telemetered navigation and altimeter data, MTRACS data). A re-examination of the filter coordinate system problem during this study has led to the conclusion that for the P2457 update, it is now preferable to continue performing the digital filtering in the test rectangular cartesian coordinate system. This will greatly simplify the combining of the pulse radar data with data from other systems (see Section 5).

4.2.8 Output Characteristics

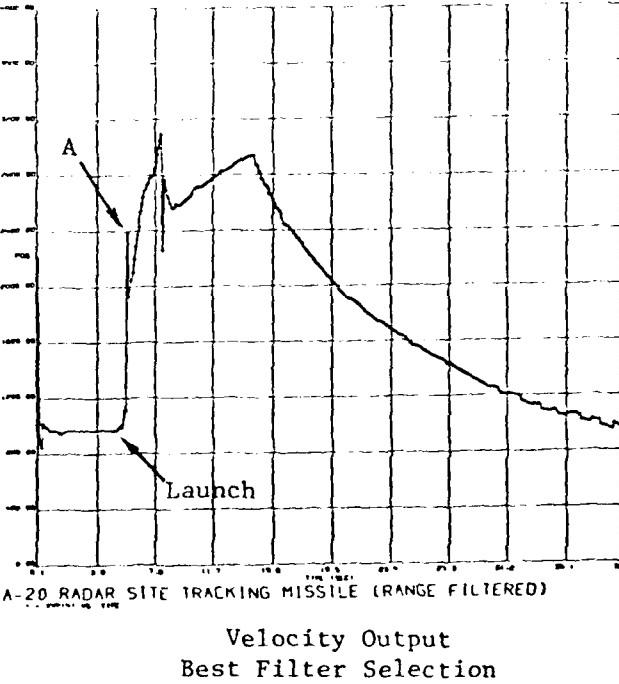
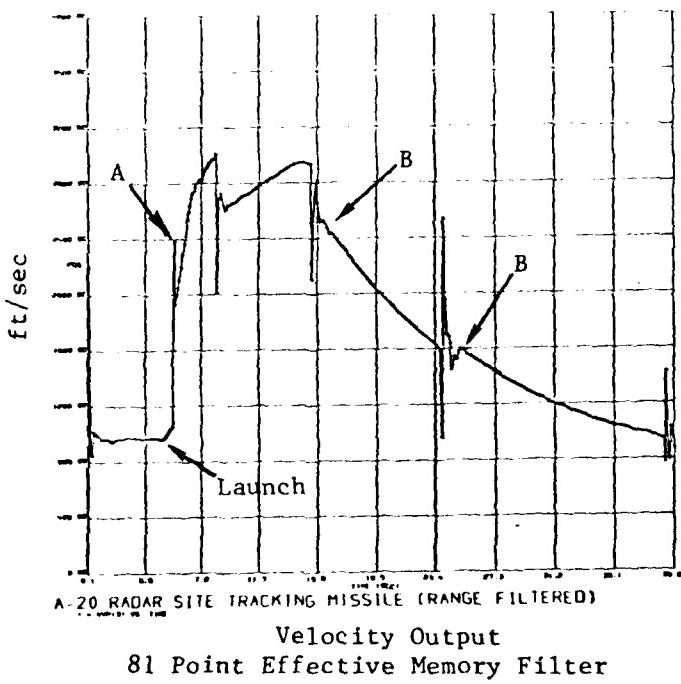
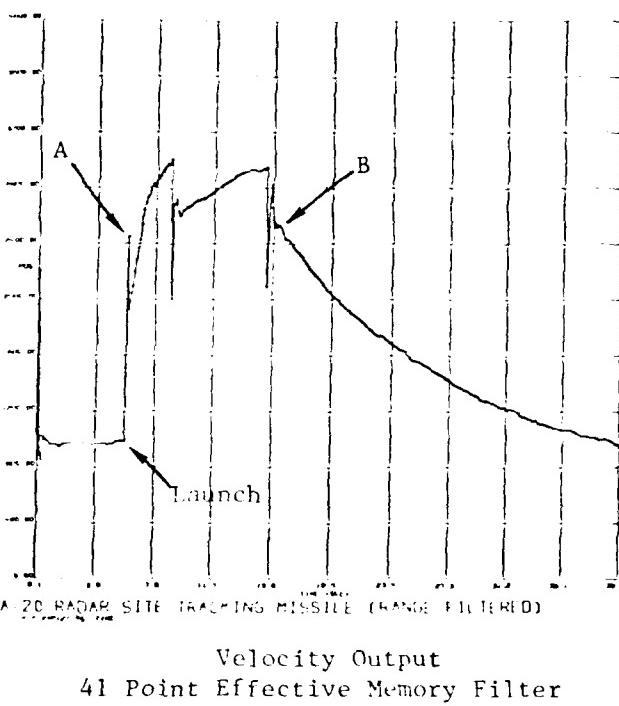
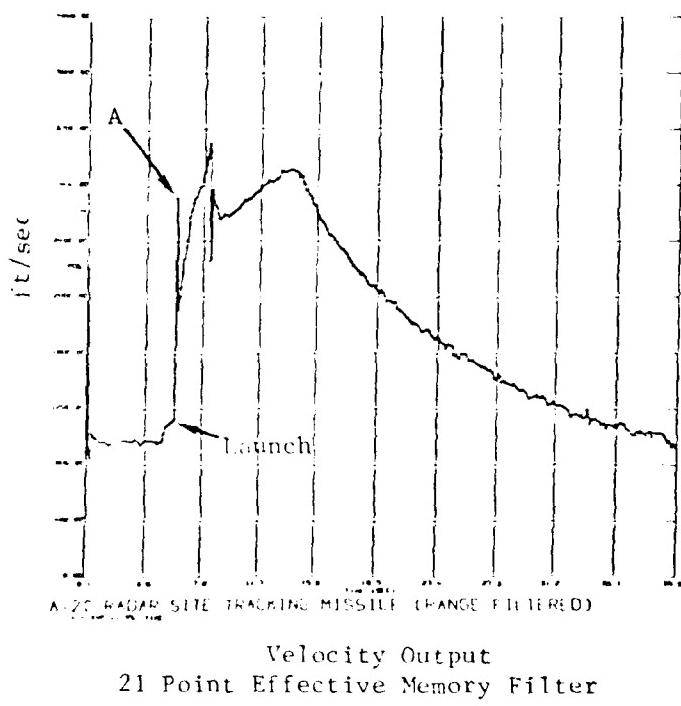
Examples of the output characteristics of the proposed adaptive exponential filter are shown in Figures 4.2.8.1 and 4.2.8.2. The input data was Site A-20 radar range data derived from a simulated AMRAAM trajectory and realistically contaminated with noise. Figure 4.2.8.1 contains plots of the filter velocity outputs in feet/second versus time in seconds. Figure 4.2.8.2 contains plots of the filter residual outputs in feet versus time. The effective memories of the shortest, intermediate and longest filters were 21, 41 and 81 points, respectively. The plots of the outputs which the adaptive process determined best at each time point are in the lower right hand corner of each figure. Some of the characteristics of interest in the two figures were the following:

1. Because the missile attained maximum acceleration within a fraction of a second after ignition, causing high order derivatives to appear in the signal, all three of the filters lagged to such an extent that the adaptive logic required them to reinitialize during the first two seconds of flight (event marked A in the figures).
2. The 41 and 81 point filters diverged at points where rapid changes in missile dynamics occurred. The divergence was detected by the adaptive process through the trended residual test (see Section 4.2.6).

The divergence can easily be seen in the residual plots of Figure 4.2.8.2 at the points marked B.

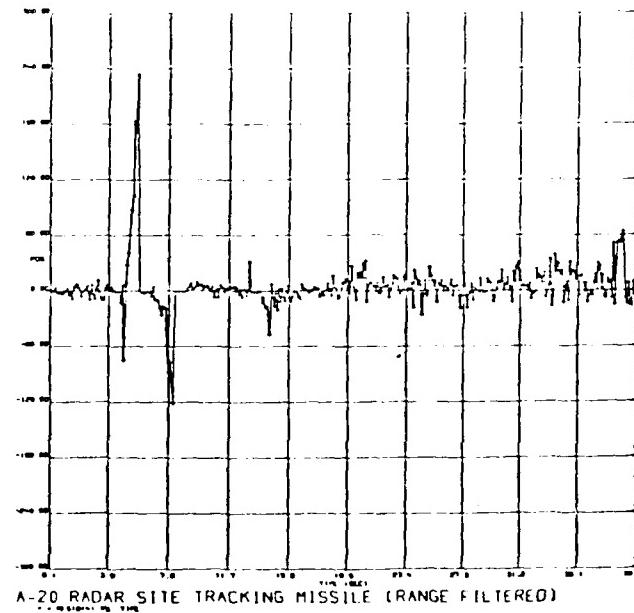
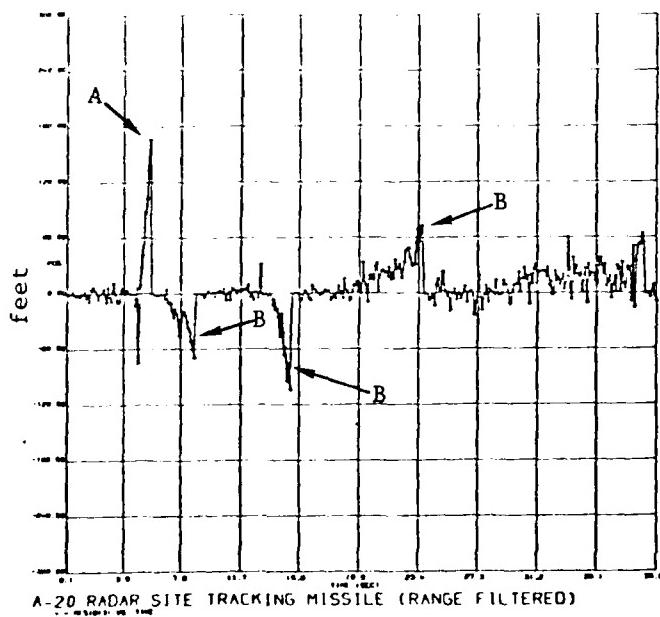
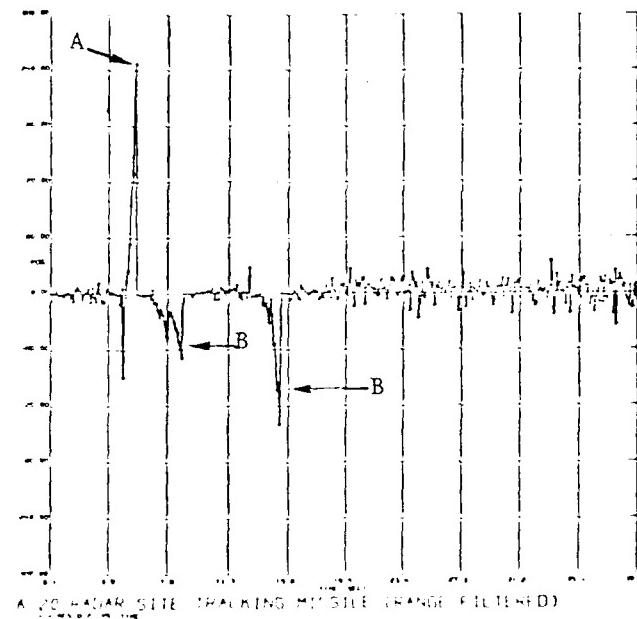
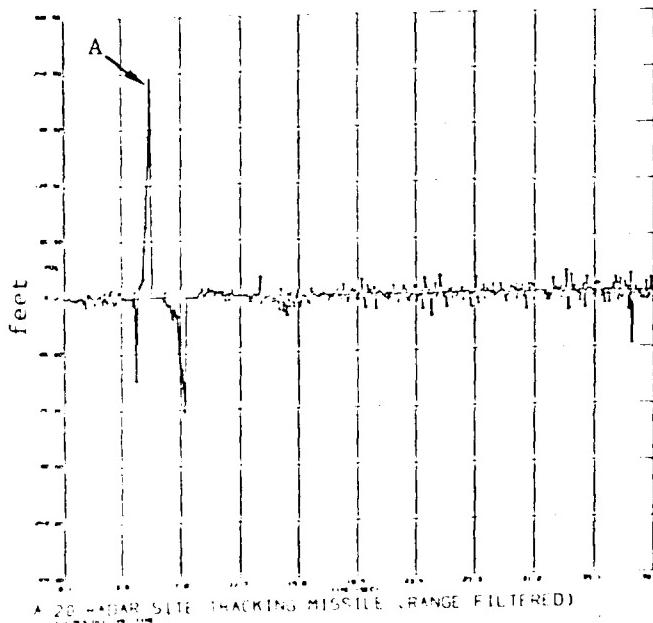
The effect on the velocity outputs can also be seen at the points marked B in Figure 4.2.8.1.

3. The effectiveness of the multiple filter adaptive process can be seen in the "Best Filter" output plot of Figure 4.2.8.1. Note that the adaptive process tended to select the shortest memory filter output during periods of high dynamics and the smooth longer memory filter outputs during periods of low dynamics.



Velocity Outputs of the Adaptive Exponential Filter
Simulated A-20 Range Data on AMRAAM Test

Figure 4.2.8.1



Residual Outputs of the Adaptive Exponential Filter
Simulated A-20 Range Data on AMRAAM Test

Figure 4.2.8.2

5.0 REAL-TIME TELEMETRY DATA PROCESSING ALGORITHMS

5.1 Background

Many of the advanced weapon programs planned for development by AD will require low level tests in which the acquired radar data will be severely degraded due to clutter, multipath and refraction effects. Because of the high dynamics of the vehicles, radar track will be difficult to reacquire if it is lost. An alternative to the use of only radar data for range safety control of advanced weapon tests is the use of telemetered on-board sensor data as back-up to the radar data or in a combined radar/telemetry (TM) data trajectory solution. On-board sensor data which might be used to augment the tracking radar data include:

- Inertial navigation system (INS) data
- Radio navigation system (RNS) data
- Radar or barometric altimeter data
- Status information (e.g., seeker lock-on indication, loss of command/control signal)

One of the Task SER 82-1 conclusions was that: "Algorithms for the use of telemetered inertial navigation and altimeter data should be included in the P2457 update". This conclusion was based largely upon the analyses conducted during Task SER 81-3 (Reference 11). One of the conclusions resulting from 81-3 was the following:

Vertical coordinate and IIP data can be improved in quality by an order of magnitude (and sometimes more)

by judiciously combining the radar data with telemetered altimeter data and INS position and velocity data. The combinative methods developed during this study are both feasible and desirable for the real-time estimation of present position coordinates and IIPs for use in the range safety control of AD weapon tests. The combinative methods, involving the application of least squares exponential filters to radar-INS data differences and to radar-altimeter data weighted averages, are described in this report (Reference 11).

An example of the improvement in IIP quality which can be achieved by combining radar and TM data is shown in Figures 5.1.1 through 5.1.4, the results of a realistic simulation of data acquired on a low-level test over the Gulf. Figure 5.1.1 shows the noisy, but unbiased, IIP error trace resulting from the use of radar data only. Figure 5.1.2 exhibits a typical smooth, but trended, IIP error trace resulting from INS data only. Figure 5.1.3 shows that, by combining the radar and INS data, a relatively smooth, untrended IIP trace results. Similarly, Figure 5.1.4 shows that by combining low elevation radar data with altimeter data, the IIP trace can be greatly improved.

A purpose of this study was to examine existing radar/TM combinative algorithms, develop new ones if necessary, and document the recommended algorithms in a form conducive to efficient incorporation in a P2457 update.

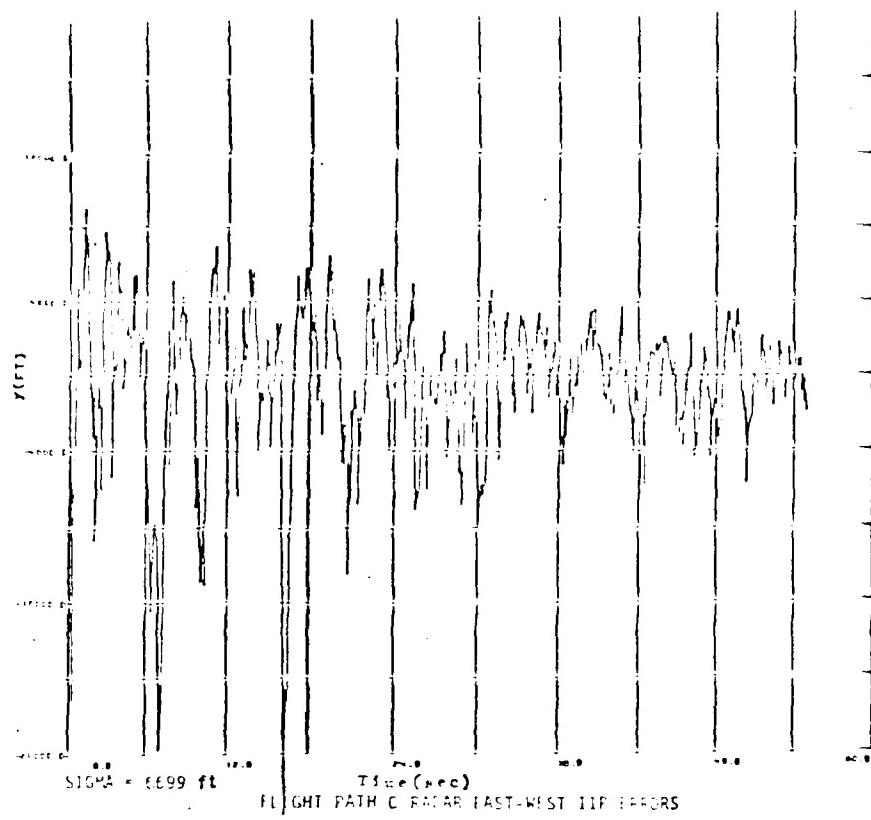


Figure 5.1.1

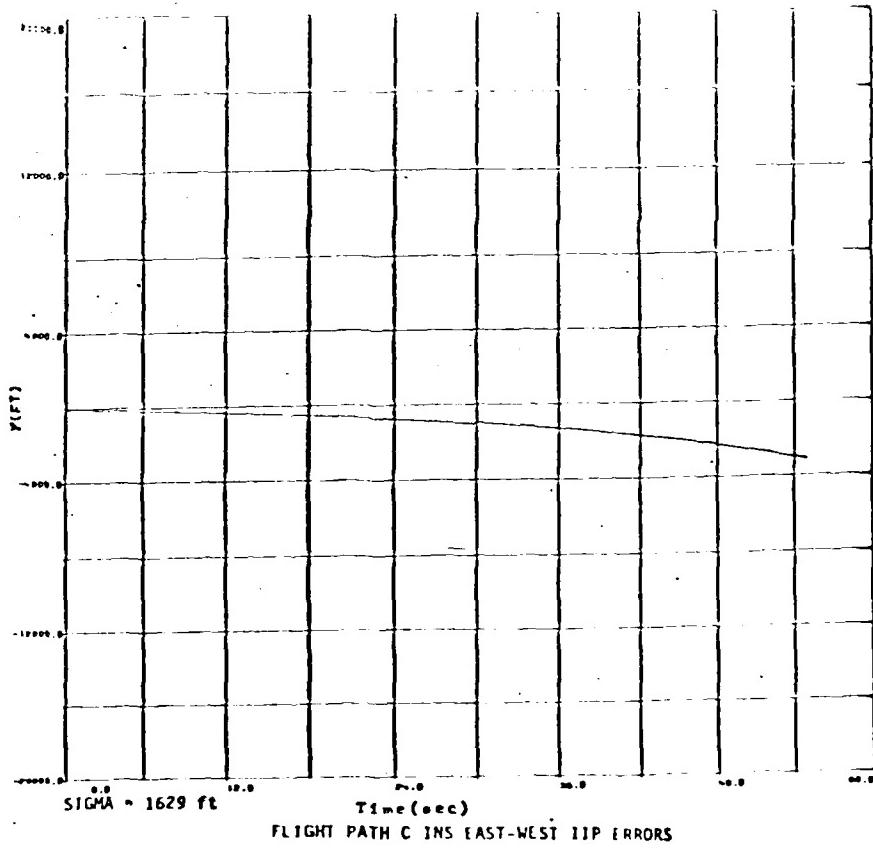


Figure 5.1.2
5-3

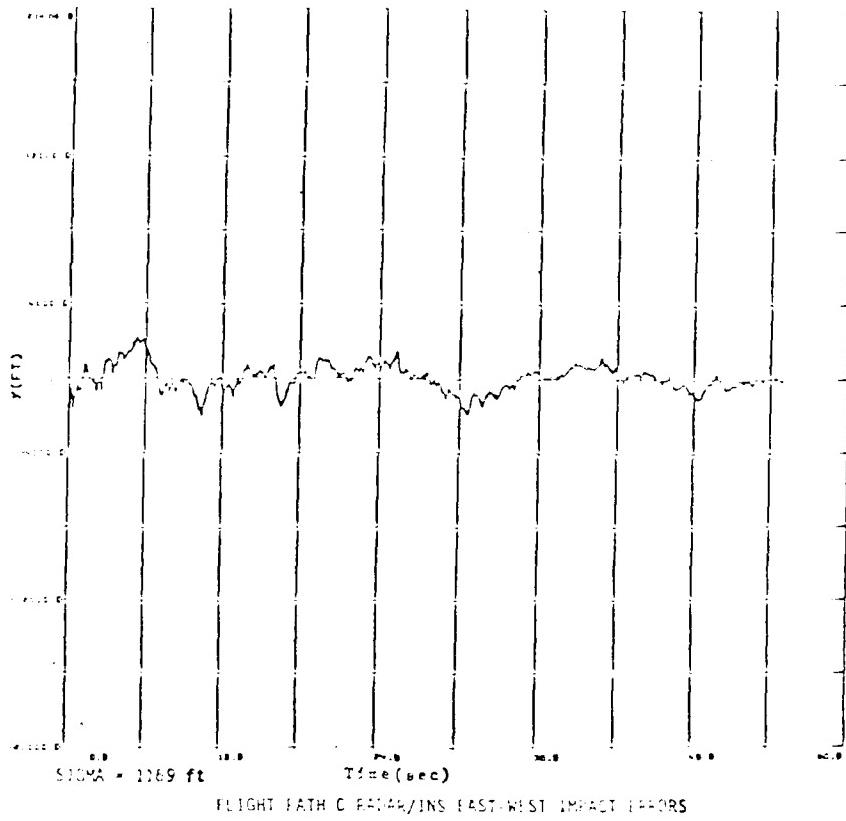


Figure 5.1.3

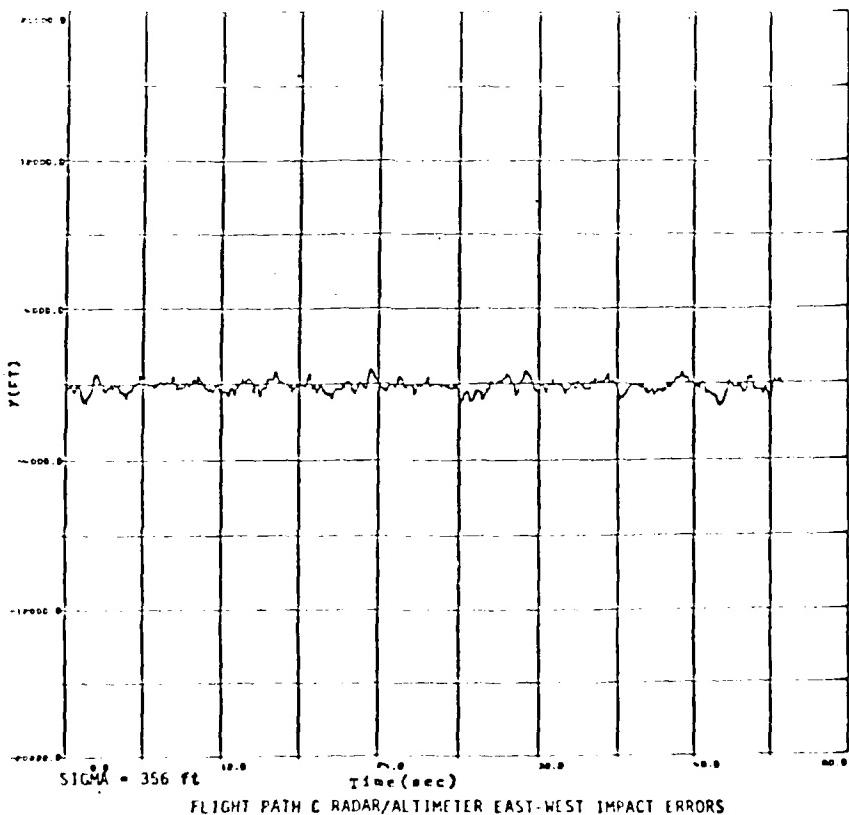


Figure 5.1.4

5.2 Coordinate Systems

There is an enormously wide variation in the types of coordinate systems in which telemetered inertial navigation and guidance system data have been presented. Variations include the following types of coordinates:

- Geodetic latitude, longitude, altitude
- Rectangular cartesian
- Scrunched (non-orthogonal axes)
- Earth fixed
- Inertial
- Topocentric
- Earth centered
- Flight line oriented axes
- North-south, east-west oriented axes
- Tilted axes

A common additional problem in such data is that the position data may be degraded by telemetering an inadequate number of bits to provide a smooth trajectory plot (e.g., the least bit telemetered by one aircraft INS is .001° in latitude and longitude, or as much as 365 ft.) This deficiency can often be overcome by numerically integrating the telemetered velocity data, for which the least bit is normally less than one ft/sec.

An in-depth analysis of all of the coordinate system, least bit and other problems involved in transforming telemetered

INS data into a form suitable for use in range safety control was considered to be beyond the scope of this study. These problems must be considered on an individual weapon system basis and a suitable transformation subroutine developed when a requirement for real-time use in range safety control is levied. It will be assumed here that the telemetered data has been transformed into a form suitable for input to the XYZLLH subroutine.

5.3 Recommended Radar/INS Combinative Algorithm

It was concluded in Task SER 81-3 (Reference 11) that a Kalman or batch filter "best estimate of trajectory (BET)" type approach to combining radar and TM data in real-time was too time consuming for use on the multiple vehicle test scenarios planned for the Eglin ranges. A re-examination of the problem during this study has reaffirmed this conclusion. It is recommended that the algorithms developed in Task 81-3, modified to utilize the filter developed during this study, be included in the P2457 update. The complete algorithm can be expressed as follows:

$$X_B = X_{INS} + \overline{(X_R - X_{INS})} \quad (5.3.1)$$

where

X_B = Best estimate of the position/velocity/acceleration vector, using radar/INS combination

X_{INS} = Vector of inertial navigation indicated coordinates

X_R = Vector of radar-indicated coordinates

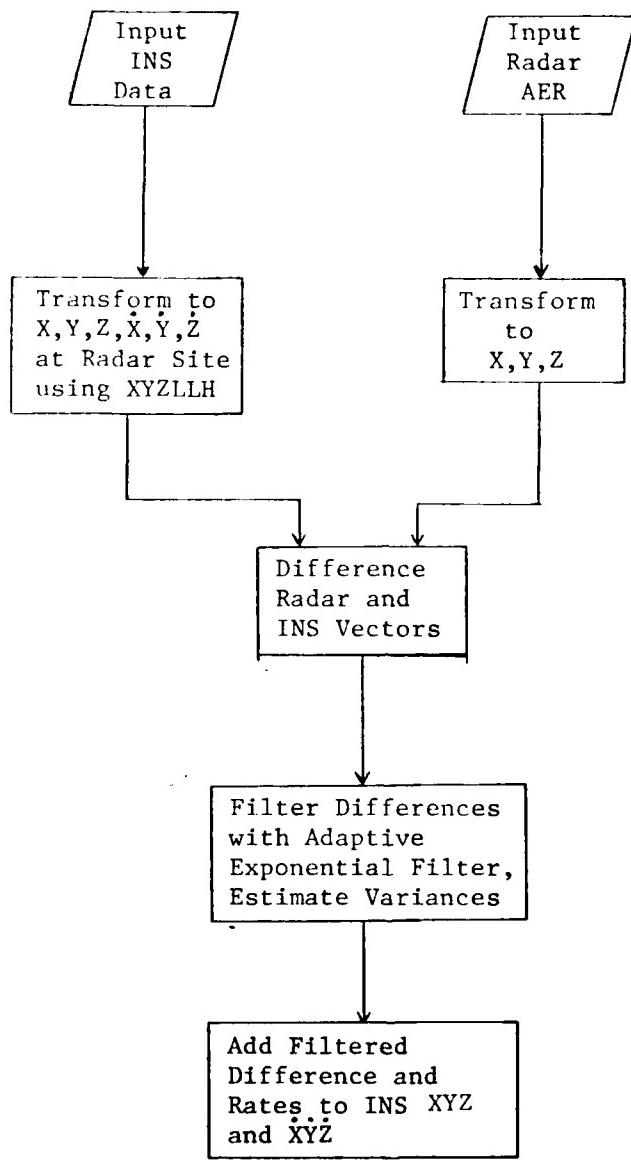
Here the bar represents low pass filtering of the radar-INS differences. It is recommended that the adaptive exponential filter developed in Section 4 of this report be used to perform the required filtering. Because the filtering process of Equation 5.3.1 is, in essence, a method for estimating low frequency errors in the INS data (e.g., the quadratic error trend of Figure 5.1.2), the required effective memory inputs to the filter will be much longer than for radar data alone. A minimum effective memory of 10 seconds is suggested.

A conceptual flow chart of the combinative radar/INS data processing is shown in Figure 5.3.1.

5.4 Recommended Radar/Altimeter Combinative Algorithm

Since radar or barometric altimeters supply only one of the three coordinates required to define a present position vector, a different combinative algorithm than that used in combining radar and INS data is required. The combinative algorithm recommended here is a modified version of the algorithm proposed in Reference 11, the only major modification being in the estimation of the weights to be used in the weighted average process. The weighted average algorithm used to obtain a best estimate of the Y (up) coordinates is as follows:

$$Y_B = \left(\sigma_Y^{-2} + \sigma_A^{-2} \right)^{-1} (Y_R \sigma_Y^{-2} + Y_A \sigma_A^{-2}) \quad (5.4.1)$$



Conceptual Flow of Radar/INS Combinative Data Process

Figure 5.3.1

where

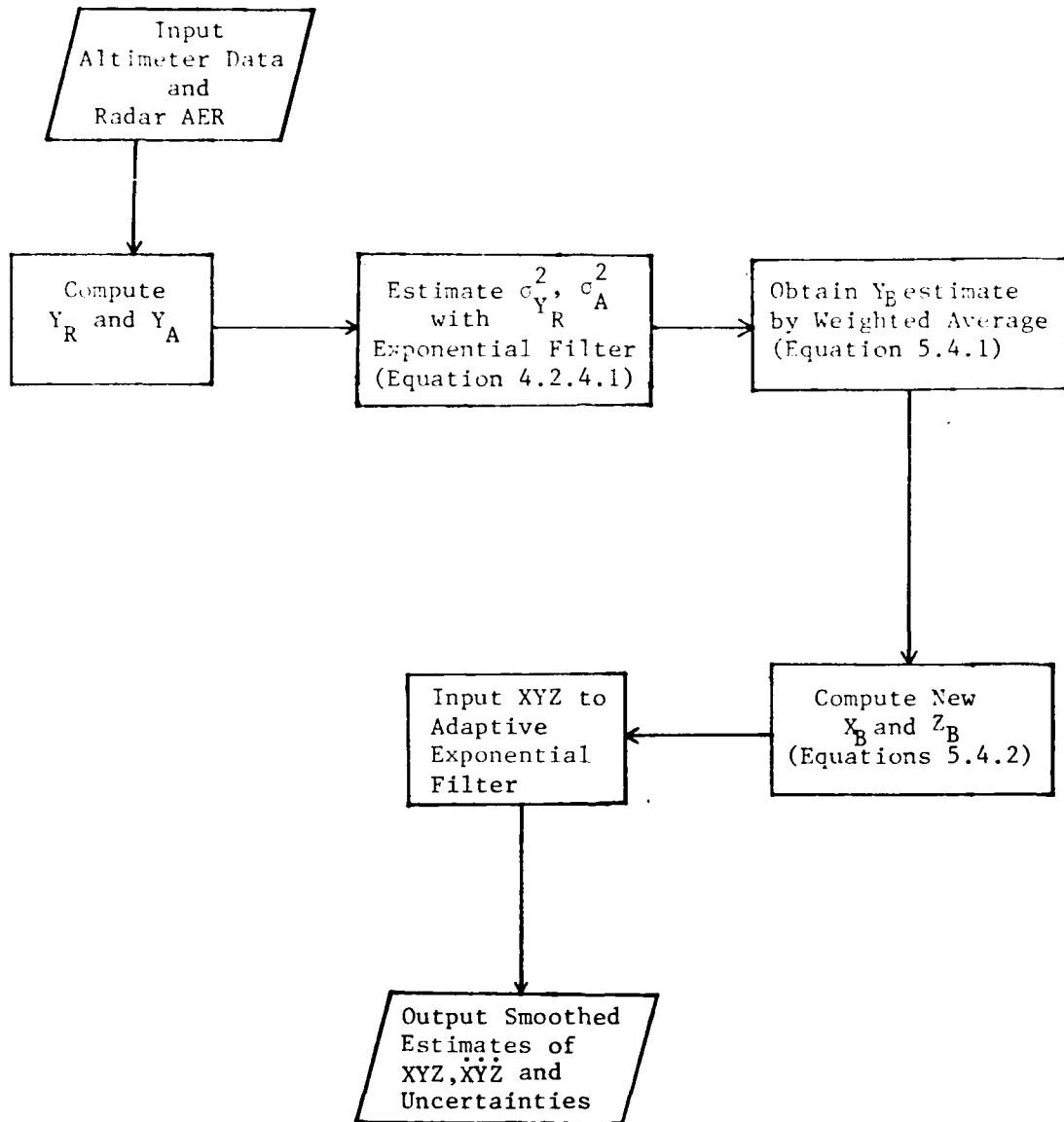
- Y_B = weighted least squares estimate of Up coordinate relative to the radar
 Y_R = radar indicated Y , corrected for refraction
 $Y_A \approx H_A - H_R - R^2 \cos^2 E / 2R_e$, a good approximation of altimeter indicated Y for AD weapon tests
 H_R = altitude of radar site
 H_A = altimeter indicated altitude above sea level
 R_e = mean radius of earth

Rather than using a priori values for $\sigma_{Y_R}^2$ and σ_A^2 , as was suggested in Reference 11, it is recommended that these variances be estimated by passing Y_R and H_A through the exponential filter and obtaining variance estimates by Equations 4.2.4.1. After obtaining the new estimate of Y_B using Equation 5.4.1, improved estimates of X and Z can be computed as follows:

$$X_B = (R^2 - Y^2)^{1/2} \cos A \quad (5.4.2)$$
$$Z_B = (R^2 - Y^2)^{1/2} \sin A$$

where A is the radar azimuth measurement and R is the refraction corrected radar range.

After obtaining the improved estimates of X , Y , Z using the combinative algorithm, the data should be input to the exponential filter to obtain a smoothed state vector and the related uncertainty estimates. A conceptual flow chart of the radar/altimeter data combinative process is shown in Figure 5.4.1.



Conceptual Flow of Combinative Radar/Altimeter Data Processing

Figure 5.4.1

REFERENCES

1. Real-Time Range Safety Algorithm Requirements (Engineering Report ER-TC-SER 82-1, Tybrin Corporation, 7 May 1982).
2. Impact Predictors for Range Safety at ADTC (Engineering Report ER-E00-014, Science Applications, Inc., 10 June 1975).
3. Impact Point Estimation: Fragments with Large Ballistic Coefficients (Engineering Report ER-E00-030, Science Applications, Inc., 30 June 1976).
4. Debris Centerline Development: Debris Triangle (Engineering Report ER-E00-018, Science Applications, Inc., 12 June 1975).
5. Determination of Drag Characteristics of Terminated Vehicle Fragments (Engineering Report ER-TC-006, Tybrin Corporation, 29 February 1980).
6. Debris Triangle Error Propagation (Engineering Report ER-E00-U29, Science Applications, Inc., 30 June 1976).
7. Land Range Impact Prediction - P1873, Program Source Listing, September 1974.
8. Radar Digital Filters for Real-Time Range Safety Usage (Letter Report LR-E00-003, Science Applications, Inc., 9 October 1974).
9. An Effective Radar Digital Filter for Range Safety Applications (Engineering Report ER-E00-017, Science Applications, Inc., 25 June 1975).
10. Introduction to Sequential Smoothing and Prediction (Norman Morrison, Ph.D., MaGraw Hill Book Co., 1969).
11. Feasibility/Desirability Study for Real-Time Use of TM Data for Range Safety Control (Engineering Report ER-TC-SESR 81-3, Tybrin Corporation, 26 June 1981).
12. Tactical Aircraft Overpressure Signature Prediction (ER-TC-SESR 80-1, Tybrin Corporation, 12 August 1980).
13. The Programmer's Handbook (Edited by R.E. Fundolic, Report No. K-1729, Union Carbide Corp., Oak Ridge, TN, 9 February 1968).
14. RPV Risk Assessment (Engineering Report ER-TC-SESR 80-3, Part 1, Tybrin Corporation, 2 September 1980).
15. Compilation of Papers Presented at the On-Axis Software Conference (ETR-TR-74-01, September 1974).

APPENDIX A
LAGRAN Interpolation/Extrapolation Subroutines

Subroutine LAGRAN performs three variable, 2nd degree polynomial, Lagrangian interpolation of tables of the form shown in Figure A.1. A FORTRAN listing of the subroutine is presented on Pages A-3 through A-5. The elements of the calling sequence are as follows:

XX is the X coordinate of the desired interpolated or extrapolated point,
YY is the Y coordinate of the desired interpolated or extrapolated point,
ZZ is the Z coordinate of the desired interpolated or extrapolated point,
X is the array of X coordinates,
Y is the array of Y coordinates,
Z is the array of Z coordinates,
TAB is the three dimensional array of function values, given by TAB (I,J,K) = F(X(I), Y(J), Z(K)),
NX is the integer number of elements in the X array,
NY is the integer number of elements in the Y array,
NZ is the integer number of elements in the Z array,
M is 3 for second degree polynomial interpolation,
IEXTX is an integer returned by the subroutine which has the following meanings:
IEXTX = 1 if extrapolation occurred above X table
IEXTX = 0 if interpolation occurred
IEXTX = -1 if extrapolation occurred below X table
IEXTX = M if M>NY
IEXTY is the same for the Y array as IEXTX,
ANS is the returned interpolated or extrapolated function approximation.

Example of
Three-Dimensional Table Input
to LAGRAN

POSITION OF ELEVATOR AS A FUNCTION OF ANGLE OF ATTACK (PARTIAL), ALTITUDE (METERS), AND WING NUMBER.

1980-08-14 120K - 17-5A

MARCH 1919

845-3 = 8775PK = .2294-

Digitized by srujanika@gmail.com

3451 E OF 8718 N = .24435

ՀԱՐՄԱՆ

43-5 DE ALIAS = 122925

MACH NUMBER

ALTITUDE	.400	.700	.800	.900	.450	1.100	1.300	1.500	1.700	2.000
-3	.770	.774	.776	.776	.776	.776	.776	.776	.776	.776
3000	.770	.774	.776	.776	.776	.776	.776	.776	.776	.776
5000	.810	.776	.794	.784	.794	.794	.824	.834	.814	.524
7000	.810	.812	.812	.812	.812	.812	.862	.862	.862	.862
121920	.824	.812	.912	.824	.824	.844	.844	.804	.504	.610
152400	.824	.812	.912	.824	.824	.844	.844	.804	.504	.630

Figure A,1

LAGRAN Subroutine FORTRAN LISTING

```
SUBROUTINE LAGRAN(XX,YY,ZZ,X,Y,Z,TAB,NX,NY,NZ,M,IEXTX,IEXTY,ANS)
C
C THREE VARIABLE LAGRANGIAN INTERPOLATION
C
      DIMENSION TAB(NX,NY,NZ),X(1),Y(1),Z(1),XLAG(100),DLAG(3)
      LOGICAL QUITX,QUITY
      N=NZ - 1
      DO 29 LL=2,N
         IF(ZZ - Z(LL+1))30,29,29
29      CONTINUE
      LL=NZ - 1
C
C INITIALIZE
C
30      QUITX=.FALSE.
      QUITY=.FALSE.
      M1=M
      IEXTX=0
      IEXTY=0
C
C FIND THE RANGE OF INTERPOLATION ALONG X
C
      IF(M1 .LE. NX)GO TO 1
      M1=NX
      IEXTX=M1
1      DO 2 I=1,NX
         IF(XX - X(I))3,9,2
2      CONTINUE
      IEXTX=1
      GO TO 8
3      MIDX=I
      IF(MIDX .GT. 1)GO TO 4
      IEXTX=-1
      GO TO 5
4      IF(ABS(XX - X(MIDX)) .GE. ABS(XX - X(MIDX - 1)))GO TO 5
      ISXPT=MIDX - (M1 / 2)
      GO TO 6
5      ISXPT=MIDX - ((M1 + 1) / 2)
6      IF(ISXPT .GT. 0)GO TO 7
      ISXPT=1
7      IEXPT=ISXPT + M1 - 1
      IF(IEXPT .LE. NX)GO TO 10
8      ISXPT=NX - M1 + 1
      IEXPT=NX
      GO TO 10
9      QUITX=.TRUE.
C
```

LAGRAN Subroutine FORTRAN LISTING

```

C FIND THE RANGE OF INTERPOLATION ALONG Y
C
10   IF(M1 .LE. NY)GO TO 11
      M1=NY
      IEXTY=M1
11   DO 12 J=1,NY
      IF(YY - Y(J))13,19,12
12   CONTINUE
      IEXTY=1
      GO TO 18
13   MIDY=J
      IF(MIDY .GT. 1)GO TO 14
      IEXTY=-1
      GO TO 15
14   IF(ABS(YY - Y(MIDY)) .GE. ABS(YY - Y(MIDY - 1)))GO TO 15
      ISYPT=MIDY - (M1 / 2)
      GO TO 16
15   ISYPT=MIDY - ((M1 + 1) / 2)
16   IF(ISYPT .GT. 0)GO TO 17
      ISYPT=1
17   IEYPT=ISYPT + M1 - 1
      IF(IEYPT .LE. NY)GO TO 20
18   ISYPT=NY - M1 + 1
      IEYPT=NY
      GO TO 20
19   QUITY=.TRUE.
20   N=0
      DO 28 LLL=(LL - 1),(LL + 1)
          N=N + 1
          IF(QUITX .AND. QUITY)GO TO 27
          IF(QUITX .AND. .NOT. QUITY)GO TO 24
          IF(.NOT. QUITX .AND. QUITY)GO TO 23
C
C INTERPOLATE ALONG X
C
21   DO 22 K=ISYPT,IEYPT
      CALL INTER(XX,TAB(1,K,LLL),X,ISXPT,IEXPT,XLAG(K))
      GO TO 26
23   CALL INTER(XX,TAB(1,J,LLL),X,ISXPT,IEXPT,XLAG(J))
      DLAG(N)=XLAG(J)
      GO TO 28
C

```

LAGRAN Subroutine FORTRAN LISTING

```

C INTERPOLATE ALONG Y
C
24    DO 25 L=ISYPT,IEYPT
25      XLAG(L)=TAB(I,L,LLL)
26      CALL INTER(YY,XLAG,Y,ISYPT,IEYPT,YLAG)
      DLAG(N)=YLAG
      GO TO 28
27      DLAG(N)=TAB(I,J,LLL)
28      CONTINUE
C
C INTERPOLATE ALONG Z
C
B1=Z(LL - 1) - Z(LL)
B2=Z(LL - 1) - Z(LL + 1)
B3=Z(LL) - Z(LL + 1)
A1=ZZ - Z(LL - 1)
A2=ZZ - Z(LL)
A3=ZZ - Z(LL + 1)
C1=DLAG(1) / (B1 * B2)
C2=-DLAG(2) / (B1 * B3)
C3=DLAG(3) / (B2 * B3)
ANS=A2 * A3 * C1 + A1 * A3 * C2 + A1 * A2 * C3
RETURN
END
}

```

```

SUBROUTINE INTER(PT,FUN,ORD,ISPT,IEPT,ALAG)
C
DIMENSION FUN(1)-ORD(1)
SUM=0.
PROD=1.
DO 3 J=ISPT,IEPT
  PROD=PROD * (PT - ORD(J))
  DENOM=1.
  DO 2 K=ISPT,IEPT
    IF(K .NE. J)GO TO 1
    D=PT
    GO TO 2
  1   D=ORD(J)
  2   DENOM=DENOM * (D - ORD(K))
  3   SUM=SUM + FUN(J) / DENOM
ALAG=SUM * PROD
RETURN
END

```

APPENDIX B

Parachute Impact Predictor

In estimating the effects of wind on the parachute impact prediction, it is assumed that the parachute becomes imbedded in the wind at the time the chute opens. An additional assumption is that it falls at the average terminal velocity computed over each weather deck sample interval by:

$$\bar{v}_{T_i}^2 = \left(\left(\frac{2\beta}{c_i + 1} \right) + \left(\frac{2\beta}{c_{i+1}} \right) \right) / 2$$

where

\bar{v}_{T_i} = average terminal velocity between altitudes h_i and h_{i+1}

$$\beta = \frac{W}{C_D S}$$

ρ_i = atmospheric density at ith altitude (slugs/ft³)

W = weight of parachute and load (pounds)

C_D = coefficient of drag

S = effective drag area of parachute

The horizontal distances which the parachute falls between altitudes h_{i+1} and h_i are given by:

$$\Delta X_i (\text{North}) = \Delta t_i * \left(\frac{WN_i + WN_{i+1}}{2} \right) \quad (B.2)$$

$$\Delta Z_i (\text{East}) = \Delta t_i * \left(\frac{WE_i + WE_{i+1}}{2} \right)$$

where

$$\Delta t_i = \left(h_{i+1} - h_i \right) / \bar{v}_{T_i}$$

$$WN_i = -WS_i * \cos(WD_i)$$

$$WE_i = -WS_i * \sin(WD_i)$$

WS_i = wind speed at altitude h_i

WD_i = wind direction at altitude h_i

The components ΔX_i , ΔZ_i are summed over all intervals between the ground and the altitude at which the parachute opens to give the total northward and eastward drifts of the parachute at impact. Subroutine XYZLLH can then be entered to determine the impact point latitude and longitudes. A FORTRAN listing of a version of the above parachute impact predictor, which is included in the DENG1 program (Reference 14), is presented on Pages B-3 and B-4. The linear interpolation subroutine called by the predictor is listed on Page B-5.

FORTRAN LISTING OF PARACHUTE
IMPACT PREDICTOR (PIP)

```

SUBROUTINE PIP(ICOL,JPIP)
C-- PARACHUTE IMPACT PREDICTOR -----
C      P(2,ICOL) IS X NORTH, FT.
C      P(3,ICOL) IS Y UP, FT.
C      P(4,ICOL) IS Z EAST, FT.
C      P(11,ICOL) IS ALTITUDE, FT.
      REAL LAMORI
      COMMON /TABLES/ WEATHN(50,5),WEATHER(5,50),AREAN,IWEATH,ICD,P(11,7
+),E1,E2,IFLAG,DT,PHI,RLAMDA,ANS(10),PLAT,PLON,PSI,FL,H,CFL,SFL
      COMMON /AERO/ Q,CMACH,TMACH(15),TALPHA(10),TABDNA(100),TABCA(100),
1TABTHR(30),NALPH,NP,CN,CA,NMACH,AREFS,CAERO,TMACH1(15),TABCL(100),
2TCA(100),NMACH1,NCA,CDSVST(40),NCDP,SPINAT
      COMMON /VREFD/ AREF,WEIGHT,AMASS,CDO,CLO,CDA,CLA,LOD,ABETA,GLATLM,
+ WDOT(10),ALOSS
      COMMON /TRAJ/ PVHIST(7,50),PV(7),PVT(7),PHIORI,LAMORI,PSIORI
      COMMON /FLAGS/ NOWIND,NOLIFT,NODRAG,NOTHRST,IOUT,LCOUNT,SPINFLG,
+ JIMMY,FACTOR,IENTRY,ICHUTE,GODOWN,IWING,WINGPFS
      COMMON /BDFG/ BADFLG,EDALT,HTLIM,ALTSTP,TIPIP,TFLMAX
      DIMENSION WN(50),WE(50),VT(50),ALT(50)
      FETID = 364566.0
      DTR = 0.0174532925
      NALTS = 50
      NLCDS = NCDP * 2
      BETA = WEIGHT/CDSVST(NLCDS)
4     DO 5 J=5,10
5     P(J,ICOL) = 0.0
      DO 10 I=1,NALTS
      NWTH = I-1
      IF(WEATHER(2,I).EQ.0.0) GO TO 12
      WD = WEATHER(3,I)*DTR
      WS = WEATHER(4,I)
      WN(I) = -WS * COS(WD) * 1.6877
      WE(I) = -WS * SIN(WD) * 1.6877
      D = WEATHER(2,I)
      VT(I) = SQRT(1030758.0 * BETA/D)
10    ALT(I) = WEATHER(1,I)
12    NALTS = NWTH
      Z = P(11,ICOL)
      CALL INPOL(ALT,WN,Z,WY,NALTS)
      CALL INPOL(ALT,WE,Z,WX,NALTS)
      CALL INPOL(ALT,VT,Z,VTZ,NALTS)
      SUMTIM = P(1,ICOL)
      XIP = 0.0
      YIP = 0.0
      DO 30 I=2,NALTS
      IF(ALT(I) - Z) 15,20,20
15    DALT = ALT(I) - ALT(I-1)
      DELT = 2.0 * DALT / (VT(I) + VT(I-1))
      XIP = XIP + 0.5 * DELT * (WE(I) + WE(I-1))

```

80

FORTRAN LISTING OF PARACHUTE
IMPACT PREDICTOR (PIP)

```
YIP = YIP + 0.5 * DELT * (WY + WN(I-1))
SUMTIM = SUMTIM + DELT
GO TO 40
30 CONTINUE
40 IF(SUMTIM .GT. 0. .AND. SUMTIM .LE. TFLMAX)GO TO 45
BADFLG=1.
GO TO 9000
45 P(2,ICOL) = P(2,ICOL) + YIP
P(3,ICOL) = 0.0
P(4,ICOL) = P(4,ICOL) + XIP
P(1,ICOL) = SUMTIM
P(11,ICOL) = 0.0
PHRAD = PHI0RI * DTR
COPHI = COS(PHRAD)
JPIP = 1
PARLAT = PHI0RI + (P(2,ICOL)/FETID)
PARLON = LAM0RI - (P(4,ICOL)/(FETID*COPHI))
WRITE(6,900) SUMTIM,VT(1)
900 FORMAT(1X,'IMPACT TIME (SEC) AND VELOCITY (FT/SEC)',2X,2(3X,F8.2))
WRITE(6,1000) PARLAT,PARLON
1000 FORMAT(1X,'PARACHUTE GEODETIC POSITION AT IMPACT',2(2X,F12.7))
WRITE(6,1010) (P(JJ,ICOL),JJ=2,4)
1010 FORMAT(1X,'PARACHUTE CARTESIAN POSITION AT IMPACT',3(1X,F12.1))
PIND = 1.
WRITE(10,4450) SUMTIM,PARLAT,PARLON,PIND
4450 FORMAT(1X,F12.4,2(2X,F10.7),F4.1)
9000 RETURN
END
```

FORTRAN LISTING OF LINEAR
INTERPOLATION SUBROUTINE (INPOL)

```
SUBROUTINE INPOL(X,Y,T,FT,NALTS)
DIMENSION X(50), Y(50)
IF(X(1)-T) 11,11,4
11 DO 1 I=1,NALTS
    IF(X(I)-T) 1,2,3
    1 CONTINUE
    GO TO 4
    2 FT = Y(I)
    GO TO 4
    3 K = I-1
        FT = Y(K) + (Y(I)-Y(K)) * (T -X(K)) / (X(I)-X(K))
    4 RETURN
    END
```

APPENDIX C

Adaptive Exponential Radar Data Filter Program (RADAR)

A FORTRAN listing of a test bed adaptive exponential radar data filter program (RADAR) developed during this study is presented on Page C-3 through C-7. Required inputs to the program are as follows:

Control Constants

<u>Name</u>	<u>Meaning</u>
START	Start time (seconds)
STOP	Stop time (seconds)
NMAX	Maximum number of extrapolations allowed before reinitialization occurs
RESMAX	Maximum size of residual to be used in computing noise variance estimates
SWATE	Alpha filter weight to be used in computing variance, noise variance
MAXNV	Maximum number of successive interfilter velocity differences allowed to exceed TOL2
NRMAX	Maximum number of successive residuals of some sign allowed
IGA (6 values)	Print output every IGAP point
NPOS	Word position on tape of measurement to be filtered
TSCALE	Time scale factor
TOL1	Discrepancy vector tolerance; used to determine need for extrapolation

<u>Name</u>	<u>Meaning</u>
TOL2	Tolerance on differences between velocity estimates of two filters
FILLEN (3 values)	Effective memory (points) in shortest, middle and longest filter
SPRIOR	Estimated standard deviation of low frequency noise component
NPLOTT	Plots desired: YES or NO

Tape Input: The program RADAR currently accepts only the output of the AMRAAM radar data simulation program SITE.

Examples of the printed ouput of RADAR are presented in Figures C.1 through C.6.

```

PROGRAM RADAR(INPUT,OUTPUT,TAPE2,TAPE3,TAPE4,TAPE5,TAPE6,
1 TAPE15=INPUT,TAPE10)
DIMENSION XPP(4),XVP(4),XAP(4),REPS(3),NEX(3),IFLAG(3),FWATE(3,3)
DIMENSION RESID(4),RNSIG(3),RPSIG(4),RSSIG(3,2),VRF(3,2),VRFE(4,2)
DIMENSION NP(3),PRESID(3),NRESID(3),THETA(3),IPRINT(6)
DIMENSION IGAP(6),WORD(72),FILLEN(3),KFG(12),TIT(3),LFLAG(3)
CHARACTER *3 NPLOTT
REWIND 10
NV32=0
NV21=0
DO 990 I=1,6
990   IPRINT(I)=-1001
DO 1010 JR=1,3
   RESID(JR)=0.
   PRESID(JR)=0.
   RNSIG(JR)=0.
   REPS(JR)=0.
   NRESID(JR)=0
   NP(JR)=0
   NEX(JR)=0
   IFLAG(JR)=1
   XPP(JR)=0.
   XVP(JR)=0.
   XAP(JR)=0.
DO 1010 K=1,2
   VRFE(JR,K)=0.
1010   RSSIG(JR,K)=0.
READ(15,*)START,STOP,NMAX,RESMAX,SWATE,MAXNU,NRMAX,IGAP,NPOS,
1 TSCALE
C
C DISCREPANCY AND VELOCITY TOLERANCE
C
C     READ(15,*)TOL1,TOL2
C
C INPUT FILTER LENGTHS
C
C     READ(15,*)(FILLEN(JR),JR=1,3)
DO 6 JR=1,3
   THETA(JR)=1 - 4.36 / FILLEN(JR)
   FWATE(JR,1)=1 - THETA(JR)**3
   FWATE(JR,2)=15 * (1 + THETA(JR)) * (1 - THETA(JR))**2
6    FWATE(JR,3)=100 * (1 - THETA(JR))**3
C
C PRIOR ESTIMATE OF OBSERVATION NOISE
C
C     READ(15,*)SPRIOR
C     SPRI02=SPRIOR**2
C
C INPUT PLOT DETERMINATION
C

```

```

      READ(15,*)NPLOTT
      PRINT 11,START,STOP,NMAX,RESMAX,SWATE,MAXNV,NRMAX,IGAP,NPOS,
1     TSCALE,TOL1,TOL2,((FWATE(IR,JR),JR=1,3),IR=1,3),SPRIOR,NPLOTT
11    FORMAT(1H1,60X,'INPUT CONSTANTS'//10X,'RAW START TIME',T40,F12.2/
1/10X,'RAW STOP TIME',T40,F12.2//10X,'MAX EXTRAPOLATIONS ALLOWED',
2T40,I5//10X,'MAX RESIDUAL',T40,F10.3//10X,'VARIANCE FILTER WEIGHT'
3,T40,F6.3//10X,'MAXIMUM NUMBER OF VELOCITIES',T40,I3//10X,
4'MAX RESIDUALS ALLOWED',T40,I5//
610X,'PRINT TIME INTERVAL',T40,615//
710X,'WORD POSITION TO BE FILTERED',T40,I2//
910X,'TIME SCALE FACTOR',T40,F10.4//10X,'DISCREPANCY TOLERANCE',
*T40,F10.5//10X,'VELOCITY TOLERANCE',
*T40,F10.5//10X,'FILTER WEIGHTS',T40,
13(3F10.7/T40)//10X,'OBSERVATION NOISE ESTIMATE',T40,
2F10.5//10X,'PLOTTING DETERMINATION',T40,A3/1H1)
      DO 1080 JR=1,3
      TH=THETA(JR)
      ONE=1. / (TH + 1.)**5
      RPSIG(JR)=SPRI02
      VRF(JR,1)=(1. - TH) * ONE * (19. + 24. * TH + 16. * TH**2 + 6. *
1     TH**3 + TH**4)
1080  VRF(JR,2)=(1. - TH)**3 * 50. * ONE * (49. + 50. * TH + 13. *
1     TH**2)
4     READ(10)TIME,(WORD(N),N=1,72),(KFG(N),N=1,12),(TIT(N),N=1,3)
      IF(ABS(TIME - START) .LT. .001)GO TO 7
      GO TO 4
7     IF((TIME - STOP) .GT. .001)GO TO 999
      KFLAG=KFG((NPOS - 1) / 6 + 1)
      RO=WORD(NPOS)
      BESF=0.
      DO 60 JR=1,3
      TOL3=TOL1
      TOL4=3. * SQRT(RPSIG(JR))
      IF(TOL4 .GT. TOL1)TOL3=TOL4
      XPP(JR)=XPP(JR) + .1 * XVP(JR) + .005 * XAP(JR)
      XVP(JR)=XVP(JR) + .1 * XAP(JR)
      IF(KFLAG)905,905,908
905   REPS(JR)=0.
      GO TO 52
908   IF(IFLAG(JR))910,51,910
910   IF(NP(JR) .EQ. 0)XPP(JR)=RO
51    REPS(JR)=RO - XPP(JR)
      IF(IFLAG(JR))915,920,915
915   LFLAG(JR)=1
      IF(NP(JR) .LT. 5)LFLAG(JR)=0
      IF(NP(JR) - 10)60,60,920
920   IF(ABS(REPS(JR)) - TOL3)925,925,52
925   NEX(JR)=0
      GO TO 60
52    NEX(JR)=NEX(JR) + 1

```

```

      IF(NEX(JR) - NMAX)60,60,54
54    IFLAG(JR)=1
      NEX(JR)=0
      NP(JR)=0
      GO TO 910
60    CONTINUE
      DO 80 JR=1,3
      IF(IFLAG(JR))80,70,80
70    IF(NEX(JR))80,71,80
71    XPP(JR)=XPP(JR) + FWATE(JR,1) * REPS(JR)
      XVP(JR)=XVP(JR) + FWATE(JR,2) * REPS(JR)
      XAP(JR)=XAP(JR) + FWATE(JR,3) * REPS(JR)
80    CONTINUE
      DO 100 JR=1,3
      IF(IFLAG(JR) .EQ. 0)GO TO 100
      XNP=NP(JR)
      IF((XNP .GT. 10.) .AND. (NEX(JR) .GT. 0))GO TO 95
      DEN=REPS(JR) / ((XNP + 3.) * (XNP + 2.) * (XNP + 1.))
      XPP(JR)=XPP(JR) + (9 * XNP**2 + 9 * XNP + 6) * DEN
      XVP(JR)=XVP(JR) + (360. * XNP + 180.) * DEN
      XAP(JR)=XAP(JR) + 6000. * DEN
95    VRFE(JR,1)=(9. * XNP**2 + 27. * XNP + 24.) / (XNP + 1.)**3
      VRFE(JR,2)=(19200. * XNP**2 + 74400. * XNP + 68400.) /
1           (XNP + 3.)**5
100   CONTINUE
      DO 120 JR=1,3
      RESID(JR)=R0 - XPP(JR)
      TEMP=RESID(JR)
      IF(ABS(TEMP) - RESMAX)114,114,112
112   TEMP=RESMAX
114   RNSIG(JR)=RNSIG(JR) + SWATE * (TEMP**2 - RNSIG(JR))
      RPSIG(JR)=ABS(RNSIG(JR)) + SPRIO2
120   CONTINUE
      DO 140 JR=1,3
      DO 140 K=1,2
      IF(IFLAG(JR))130,124,130
124   RSSIG(JR,K)=VRF(JR,K) * RPSIG(JR)
      GO TO 140
130   RSSIG(JR,K)=VRFE(JR,K) * RPSIG(JR)
140   CONTINUE
      DO 160 JR=1,3
      IF(IFLAG(JR))144,160,144
144   IF(VRF(JR,1) - VRFE(JR,1))148,152,152
148   NP(JR)=NP(JR) + 1
      GO TO 160
152   NP(JR)=0
      IFLAG(JR)=0
160   CONTINUE
      DO 175 JR=1,3
      IF(RESID(JR) * PRESID(JR))170,170,168

```

```

168      NRESID(JR)=NRESID(JR) + 1
        GO TO 175
170      NRESID(JR)=0
175      PRESID(JR)=RESID(JR)
        IF(LFLAG(3))200,200,176
176      IF(VRFE(3,1) - VRFE(2,1))177,177,201
177      IF(VRFE(3,1) - VRFE(1,1))181,181,210
181      DELV1=XVP(3) - XVP(2)
        IF(ABS(DELV1) - TOL2)183,182,182
182      NV32=NV32 + 1
        IF(NV32 - MAXNV)184,184,200
183      NV32=0
184      IF(NRESID(3) - NRMAX)185,185,200
186      BESF=3.
200      IF(BESF)220,201,220
201      IF(LFLAG(2))210,210,202
202      IF(VRFE(2,1) - VRFE(1,1))203,203,210
203      DELV1=XVP(2) - XVP(1)
        IF(ABS(DELV1) - TOL2)205,204,204
204      NV21=NV21 + 1
        IF(NV21 - MAXNV)206,206,210
205      NV21=0
206      IF(NRESID(2) - NRMAX)207,207,210
207      BESF=2.
        GO TO 220
210      IF(LFLAG(1))212,212,218
212      IF(LFLAG(2))213,213,215
213      BESF=3
        GO TO 220
215      BESF=2
        GO TO 220
218      BESF=1
220      TIME1=TIME * TSCALE
        IBESF=BESF
        RESID(4)=RESID(IBESF)
        RPSIG(4)=RPSIG(IBESF)
        VRFE(4,1)=VRFE(IBESF,1)
        VRFE(4,2)=VRFE(IBESF,2)
        XPP(4)=XPP(IBESF)
        XVP(4)=XVP(IBESF)
        XAP(4)=XAP(IBESF)
        DO 500 I=1,6
          IPRINT(I)=IPRINT(I) + 1
          IF(IPRINT(I) .EQ. -1000)GO TO 254
          IF(IPRINT(I) .LT. IGAP(I))GO TO 500
254      IPRINT(I)=0
          GO TO (255,265,270,275,276,277),I
255      PRINT 256,TIME1
256      FORMAT(/2X,F10.2)
          PRINT 258,(RESID(JR),JR=1,4),BESF,(NEX(JR),JR=1,3),

```

```
*      (LFLAG(JR),JR=1,3)
C      WRITE(8)TIME1,(RESID(JR),JR=1,4),BESF
258  FORMAT(2X,5E15.7,6I3)
      GO TO 500
265  WRITE(2,256)TIME1
      WRITE(2,257)(SQRT(RPSIG(JR)),JR=1,4),BESF
257  FORMAT(2X,5E15.7)
C      WRITE(9)TIME1,(SQRT(RPSIG(JR)),JR=1,4),BESF
      GO TO 500
270  WRITE(3,256)TIME1
      WRITE(3,257)((VRFE(JR,KK),JR=1,4),BESF,KK=1,2)
      GO TO 500
275  WRITE(4,256)TIME1
      WRITE(4,257)(XPP(JR),JR=1,4),BESF
      GO TO 500
276  WRITE(5,256)TIME1
      WRITE(5,257)(XVP(JR),JR=1,4),BESF
C      WRITE(11)TIME1,(XVP(JR),JR=1,4),BESF
      GO TO 500
277  WRITE(6,256)TIME1
      WRITE(6,257)(XAP(JR),JR=1,4),BESF
500  CONTINUE
      READ(10)TIME,(WORD(N),N=1,72),(KFG(N),N=1,12),(TIT(N),N=1,3)
      GO TO 7
999  IF(NPLOTT .NE. 'YES')GO TO 1110
1110 STOP
      END
```

Time	Filter 1 Residual	Filter 2 Residual	Filter 3 Residual	Best Filter Residual	Best Filter Number
20.90	- .1166851E+02	- .1357551E+02	- .1608532E+02	- .1608532E+02	.3000000E+01
20.90	- .5023252E+01	- .8394788E+01	- .1292339E+02	- .1292339E+02	.3000000E+01
21.00	- .3050656E+00	- .4302493E+01	- .6785023E+01	- .8786929E+01	.3000000E+01
21.10	- .2933181E+01	- .8094942E+01	- .1387412E+02	- .1387412E+02	.3000000E+01
21.20	- .6025795E+01	- .1367421E+02	- .2185618E+02	- .2185618E+02	.3000000E+01
21.30	.2751791E+01	- .3564519E+01	- .1217817E+02	- .1217817E+02	.3000000E+01
21.40	.1305020E+01	- .4268441E+01	- .1346946E+02	- .1305020E+01	.1000000E+01
21.50	.2383596E+01	- .1957227E+01	- .1128963E+02	.2383636E+01	.1000000E+01
21.60	.3632023E+01	.1079361E+01	- .7765589E+01	.1079361E+01	.2000000E+01
21.70	- .1972979E+01	- .5169665E+01	- .1484365E+02	- .5169665E+01	.2000000E+01
21.80	- .2037006E+01	- .5923866E+01	- .1661126E+02	- .5923866E+01	.2000000E+01
21.90	- .1458271E+01	- .5794425E+01	- .1749429E+02	- .5794425E+01	.2000000E+01
22.00	- .1136749E+00	- .4317745E+01	- .1673432E+02	- .4317745E+01	.2000000E+01
22.10	- .2283854E+00	- .4347469E+01	- .1750837E+02	- .4347469E+01	.2000000E+01
22.20	- .2582746E+00	- .4304229E+01	- .1822151E+02	- .4304229E+01	.2000000E+01
22.30	- .1484098E+01	- .5989348E+01	- .2166623E+02	- .1484098E+01	.1000000E+01

D-3 Range Measurement
 Adaptive Exponential Filter Output Example:
 Residuals

Figure C.1

Time	Filter 1 Sigma	Filter 2 Sigma	Filter 3 Sigma	Best Filter Sigma	Best Filter Number
20.80	.6496727E+01	.9753150E+01	.1635178E+02	.1635178E+02	.3000000E+01
20.90	.6494628E+01	.9701935E+01	.1055221E+02	.1055221E+02	.3000000E+01
21.00	.6176927E+01	.2425258E+01	.1043831E+02	.1043831E+02	.3000000E+01
21.10	.5007357E+01	.9338307E+01	.1071474E+02	.1071474E+02	.3000000E+01
21.20	.5008649E+01	.9705080E+01	.1184094E+02	.1184094E+02	.3000000E+01
21.30	.5840091E+01	.9406635E+01	.1186485E+02	.1186485E+02	.3000000E+01
21.40	.5642553E+01	.9141456E+01	.1198417E+02	.5642553E+01	.1000000E+01
21.50	.5477907E+01	.8830897E+01	.1193687E+02	.5477907E+01	.1000000E+01
21.60	.5369390E+01	.8520935E+01	.1169341E+02	.8520935E+01	.2000000E+01
21.70	.5204297E+01	.8330404E+01	.1194101E+02	.8330404E+01	.2000000E+01
21.80	.5047636E+01	.8185010E+01	.1232567E+02	.8185010E+01	.2000000E+01
21.90	.4883088E+01	.8040837E+01	.1275582E+02	.8040837E+01	.2000000E+01
22.00	.4709176E+01	.7837997E+01	.1307379E+02	.7837997E+01	.2000000E+01
22.10	.4541767E+01	.7645708E+01	.1343195E+02	.7645708E+01	.2000000E+01
22.20	.4380454E+01	.7460678E+01	.1382135E+02	.7460678E+01	.2000000E+01
22.30	.4242566E+01	.7367255E+01	.1444724E+02	.4242566E+01	.1000000E+01

D-3 Range Measurement

Adaptive Exponential Filter Output Example:
Standard Deviations

Figure C.2

Position Velocity	Time	VRF1 VRF1	VRF2 VRF2	VRF3 VRF3	VRF-Best VRF-Best	Best Filter Number
20.80						
.5617749E+00	.2499358E+00	.1730556E+00	.1730556E+00	.1730556E+00	.3000000E+01	
.2493438E+01	.3062371E+00	.1109790E+00	.1109790E+00	.1109790E+00	.3000000E+01	
20.90						
.5617749E+00	.2499358E+00	.1697912E+00	.1697912E+00	.1697912E+00	.3000000E+01	
.2493438E+01	.3062371E+00	.1052135E+00	.1052135E+00	.1052135E+00	.3000000E+01	
21.00						
.5617749E+00	.2499358E+00	.1666476E+00	.1666476E+00	.1666476E+00	.3000000E+01	
.2493438E+01	.3062371E+00	.9984044E-01	.9984044E-01	.9984044E-01	.3000000E+01	
21.10						
.5617749E+00	.2499358E+00	.1636183E+00	.1636183E+00	.1636183E+00	.3000000E+01	
.2493438E+01	.3062371E+00	.9482709E-01	.9482709E-01	.9482709E-01	.3000000E+01	
21.20						
.5617749E+00	.2499358E+00	.1606972E+00	.1606972E+00	.1606972E+00	.3000000E+01	
.2493438E+01	.3062371E+00	.9014381E-01	.9014381E-01	.9014381E-01	.3000000E+01	
21.30						
.5617749E+00	.2499358E+00	.1578786E+00	.1578786E+00	.1578786E+00	.3000000E+01	
.2493438E+01	.3062371E+00	.8576389E-01	.8576389E-01	.8576389E-01	.3000000E+01	
21.40						
.5617749E+00	.2499358E+00	.1551571E+00	.1551571E+00	.1551571E+00	.1000000E+01	
.2493438E+01	.3062371E+00	.8150316E-01	.8150316E-01	.8150316E-01	.1000000E+01	
21.50						
.5617749E+00	.2499358E+00	.1525278E+00	.1525278E+00	.1525278E+00	.1000000E+01	
.2493438E+01	.3062371E+00	.7781972E-01	.7781972E-01	.7781972E-01	.1000000E+01	
21.60						
.5617749E+00	.2499358E+00	.1499861E+00	.1499861E+00	.1499861E+00	.2000000E+01	
.2493438E+01	.3062371E+00	.7421371E-01	.7421371E-01	.7421371E-01	.2000000E+01	
21.70						
.5617749E+00	.2499358E+00	.1475278E+00	.1475278E+00	.1475278E+00	.2000000E+01	
.2493438E+01	.3062371E+00	.7082708E-01	.7082708E-01	.7082708E-01	.2000000E+01	
21.80						
.5617749E+00	.2499358E+00	.1451487E+00	.1451487E+00	.1451487E+00	.2000000E+01	
.2493438E+01	.3062371E+00	.6764340E-01	.6764340E-01	.6764340E-01	.2000000E+01	
21.90						
.5617749E+00	.2499358E+00	.1428452E+00	.1428452E+00	.1428452E+00	.2000000E+01	
.2493438E+01	.3062371E+00	.6464768E-01	.6464768E-01	.6464768E-01	.2000000E+01	
22.00						
.5617749E+00	.2499358E+00	.1406136E+00	.1406136E+00	.1406136E+00	.2000000E+01	
.2493438E+01	.3062371E+00	.6182627E-01	.6182627E-01	.6182627E-01	.2000000E+01	
22.10						
.5617749E+00	.2499358E+00	.1384506E+00	.1384506E+00	.1384506E+00	.2000000E+01	
.2493438E+01	.3062371E+00	.5916666E-01	.5916666E-01	.5916666E-01	.2000000E+01	

D-3 Range Measurement
Adaptive Exponential Filter Output Example:
Variance Reduction Factors
Figure C.3

Time	Filter 1 Position	Filter 2 Position	Filter 3 Position	Best Filter Position	Best Filter Number
20.80	.2120022E+06	.2120071E+06	.2120096E+06	.2120036E+06	.3000000E+01
20.90	.2119137E+06	.2119177E+06	.2119216E+06	.2119216E+06	.3000000E+01
21.00	.2118266E+06	.2118306E+06	.2118351E+06	.2118351E+06	.3000000E+01
21.10	.2117374E+06	.2117426E+06	.2117484E+06	.2117484E+06	.3000000E+01
21.20	.2116451E+06	.2116528E+06	.2116610E+06	.2116610E+06	.3000000E+01
21.30	.2115608E+06	.2115672E+06	.2115758E+06	.2115758E+06	.3000000E+01
21.40	.2114763E+06	.2114819E+06	.2114911E+06	.2114763E+06	.1000000E+01
21.50	.2113937E+06	.2113981E+06	.2114074E+06	.2113937E+06	.1000000E+01
21.60	.2113136E+06	.2113161E+06	.2113250E+06	.2113161E+06	.2000000E+01
21.70	.2112293E+06	.2112325E+06	.2112421E+06	.2112325E+06	.2000000E+01
21.80	.2111451E+06	.2111490E+06	.2111597E+06	.2111430E+06	.2000000E+01
21.90	.21106618E+06	.2110661E+06	.2110778E+06	.2110661E+06	.2000000E+01
22.00	.2109800E+06	.2109842E+06	.2109966E+06	.2109842E+06	.2000000E+01
22.10	.2108987E+06	.2109028E+06	.2109160E+06	.2109028E+06	.2000000E+01
22.20	.2108180E+06	.2108220E+06	.2108359E+06	.2108220E+06	.2000000E+01
22.30	.2107365E+06	.2107410E+06	.2107561E+06	.2107365E+06	.1000000E+01

D-3 Range Measurement
 Adaptive Exponential Filter Output Example:
 Position
 Figure C.4

Time	Filter 1 Velocity	Filter 2 Velocity	Filter 3 Velocity	Best Filter Velocity	Best Filter Number
20.80	-8675019E+03	-8618171E+03	-8600004E+03	-8600004E+03	.3000000E+01
20.90	-8715333E+03	-8576802E+03	-8530318E+03	-8530318E+03	.3000000E+01
21.00	-8853784E+03	-8515831E+03	-8467658E+03	-8467658E+03	.3000000E+01
21.10	-895140E+03	-8472693E+03	-8414180E+03	-8404180E+03	.3000000E+01
21.20	-8731584E+03	-8455950E+03	-8391949E+03	-8250949E+03	.3000000E+01
21.30	-8615548E+03	-8396056E+03	-8286136E+03	-8285136E+03	.3000000E+01
21.40	-8555264E+03	-8333913E+03	-8222703E+03	-828254E+03	.1000000E+01
21.50	-851475E+03	-8274104E+03	-8156782E+03	-8413455E+03	.1000000E+01
21.60	-8265279E+03	-8194953E+03	-8087048E+03	-8194953E+03	.2000000E+01
21.70	-8241135E+03	-8143745E+03	-8024523E+03	-8143745E+03	.2000000E+01
21.80	-8222032E+03	-8096803E+03	-7963661E+03	-8096803E+03	.2000000E+01
21.90	-8193112E+03	-8050276E+03	-793553E+03	-8050276E+03	.2000000E+01
22.00	-8135493E+03	-7998079E+03	-7842645E+03	-7998079E+03	.2000000E+01
22.10	-8060751E+03	-7946744E+03	-7782347E+03	-7946744E+03	.2000000E+01
22.20	-8027116E+03	-7895946E+03	-7722585E+03	-7895946E+03	.2000000E+01
22.30	-8002500E+03	-7853459E+03	-7665143E+03	-8002500E+03	.1000000E+01

D-3 Range Measurement
 Adaptive Exponential Filter Output Example:
 Velocity
 Figure C.5

Time	Filter 1 Acceleration	Filter 2 Acceleration	Filter 3 Acceleration	Best Filter Acceleration	Best Filter Number
20.30	.7764603E+02	.8165158E+02	.8139465E+02	.8139455E+02	.3000000E+01
20.90	.6365384E+02	.8033548E+02	.8084298E+02	.8084238E+02	.3000000E+01
21.00	.6611106E+02	.7961102E+02	.8048876E+02	.8048876E+02	.3000000E+01
21.10	.6282348E+02	.7824705E+02	.7935392E+02	.7935392E+02	.3000000E+01
21.20	.5198424E+02	.7594298E+02	.7917146E+02	.7917146E+02	.3000000E+01
21.30	.5393436E+02	.7334237E+02	.7875526E+02	.7875526E+02	.3000000E+01
21.40	.6928132E+02	.7462314E+02	.7831373E+02	.5928132E+02	.1000000E+01
21.50	.6356987E+02	.7429336E+02	.7797145E+02	.6356987E+02	.1000000E+01
21.60	.7010341E+02	.7447523E+02	.7774452E+02	.7447523E+02	.2000000E+01
21.70	.6655427E+02	.7360415E+02	.7733208E+02	.7360415E+02	.2000000E+01
21.80	.6238997E+02	.7260600E+02	.7689286E+02	.7260600E+02	.2000000E+01
21.90	.6026673E+02	.7162965E+02	.7645232E+02	.7162965E+02	.2000000E+01
22.00	.6066224E+02	.7090212E+02	.7605068E+02	.7090212E+02	.2000000E+01
22.10	.5965141F+02	.7016959E+02	.7564984E+02	.7016959E+02	.2000000E+01
22.20	.5918581E+02	.6944433E+02	.7525164E+02	.6944433E+02	.2000000E+01
22.30	.5651711E+02	.6843514E+02	.7481189E+02	.5651711E+02	.1000000E+01

D-3 Range Measurement
 Adaptive Exponential Filter Output Example:
 Acceleration
 Figure C.6

STATEMENT OF WORK

During Task SER 82-1, algorithms required for inclusion in the VAX 11/780 computer real-time range safety programs were identified. The identified algorithms were of three types: a) algorithms being utilized in the current (P2457) software, requiring no modifications, b) algorithms utilized in P2457, requiring modifications, and c) new algorithms not currently included in P2457.

Among the new algorithms recommended were the following:

- 1) Table look-up drag and wind corrected instantaneous impact predictions (IIPs) with uncertainty ellipses.
- 2) Adaptive exponential radar data filters.
- 3) Utilization of telemetered inertial navigation system (INS) and altimeter data.

The contractor is directed to develop the above three algorithms and present them in a form conducive to fast, efficient incorporation in the VAX 11/780 real-time range safety software.

The task will proceed within the following guidelines:

- 1) The algorithms will be defined both in conventional mathematical and Fortran notation.
- 2) Flow diagrams of the algorithms will be developed and presented.
- 3) The validity of the algorithms will be demonstrated by running realistic test cases on the AD CDC computer.

